FIG. 1

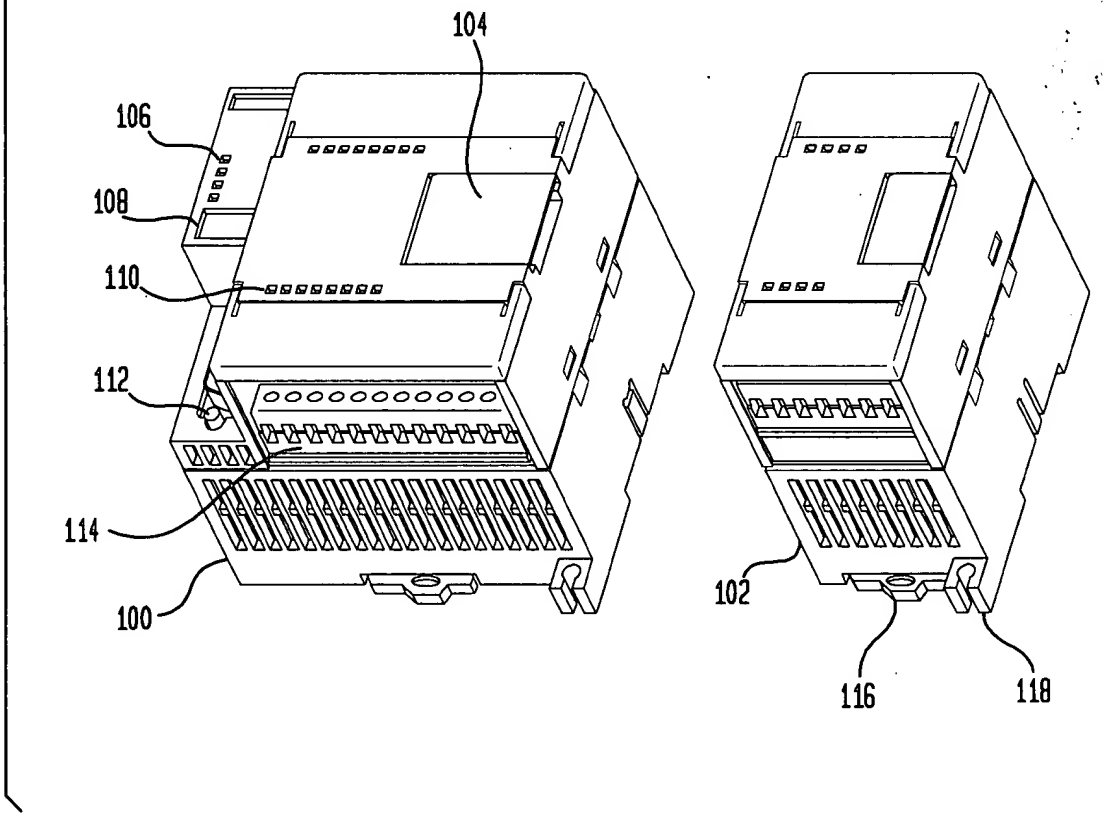


FIG. 2

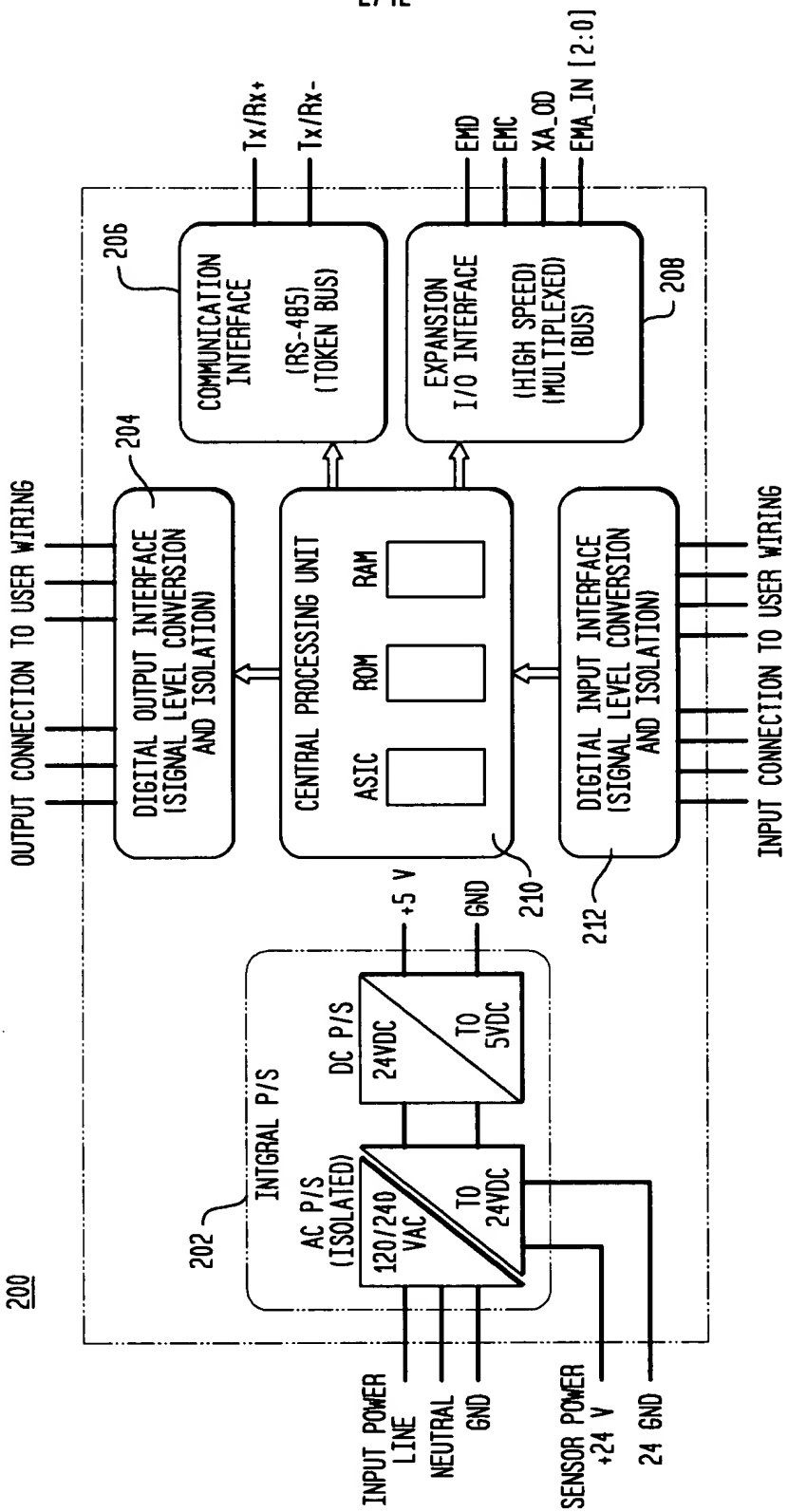

200
202
INTGRAL P/S
AC P/S (ISOLATED)
120/240 VAC
TO 24VDC
DC P/S
24VDC
TO 5VDC
+5 V
GND
INPUT POWER LINE
NEUTRAL
GND
SENSOR POWER +24 V
24 GND
OUTPUT CONNECTION TO USER WIRING
INPUT CONNECTION TO USER WIRING
204
DIGITAL OUTPUT INTERFACE (SIGNAL LEVEL CONVERSION AND ISOLATION)
CENTRAL PROCESSING UNIT
ASIC
ROM
RAM
210
212
DIGITAL INPUT INTERFACE (SIGNAL LEVEL CONVERSION AND ISOLATION)
206
COMMUNICATION INTERFACE
(RS-485)
(TOKEN BUS)
Tx/Rx+
Tx/Rx-
208
EXPANSION I/O INTERFACE
(HIGH SPEED)
(MULTIPLEXED)
(BUS)
EMD
EMC
XA_OD
EMA_IN [2:0]

FIG. 3

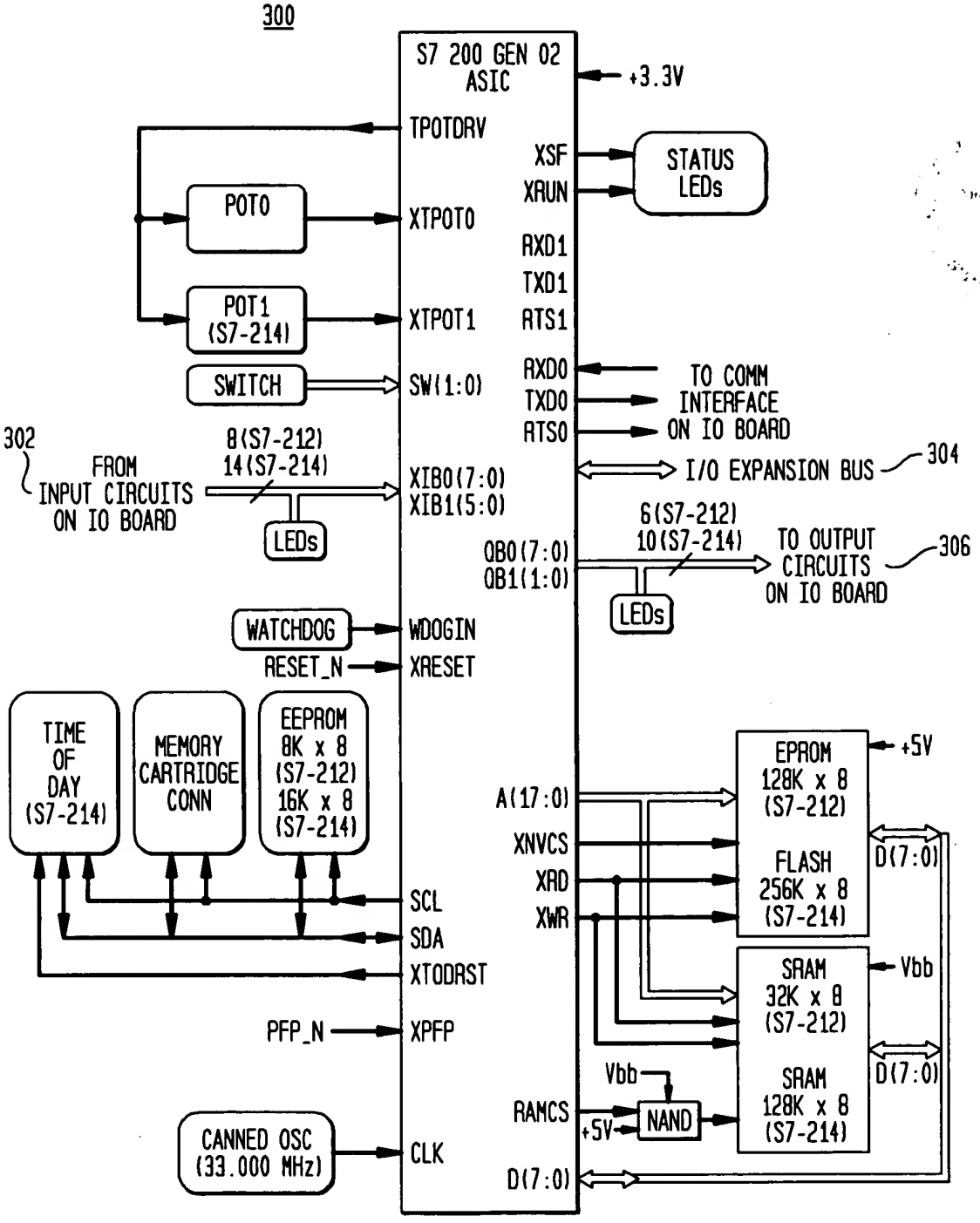

300
S7 200 GEN 02 ASIC
+3.3V
TPOTDRV
XSF
XRUN
STATUS LEDs
POT0
XTPOT0
RXD1
TXD1
POT1 (S7-214)
XTPOT1
RTS1
SWITCH
SW(1:0)
RXD0
TXD0
RTS0
TO COMM INTERFACE ON IO BOARD
302
FROM INPUT CIRCUITS ON IO BOARD
8(S7-212) 14(S7-214)
XIB0(7:0) XIB1(5:0)
I/O EXPANSION BUS
304
LEDs
6(S7-212) 10(S7-214)
QB0(7:0) QB1(1:0)
TO OUTPUT CIRCUITS ON IO BOARD
306
LEDs
WATCHDOG
WDOGIN
RESET_N
XRESET
TIME OF DAY (S7-214)
MEMORY CARTRIDGE CONN
EEPROM 8K x 8 (S7-212) 16K x 8 (S7-214)
A(17:0)
EPROM 128K x 8 (S7-212)
+5V
XNVCS
XRD
XWR
FLASH 256K x 8 (S7-214)
D(7:0)
SCL
SDA
XTODRST
SRAM 32K x 8 (S7-212)
Vbb
SRAM 128K x 8 (S7-214)
D(7:0)
PFP_N
XPFP
Vbb
RAMCS
+5V
NAND
CANNED OSC (33.000 MHz)
CLK
D(7:0)

FIG. 4

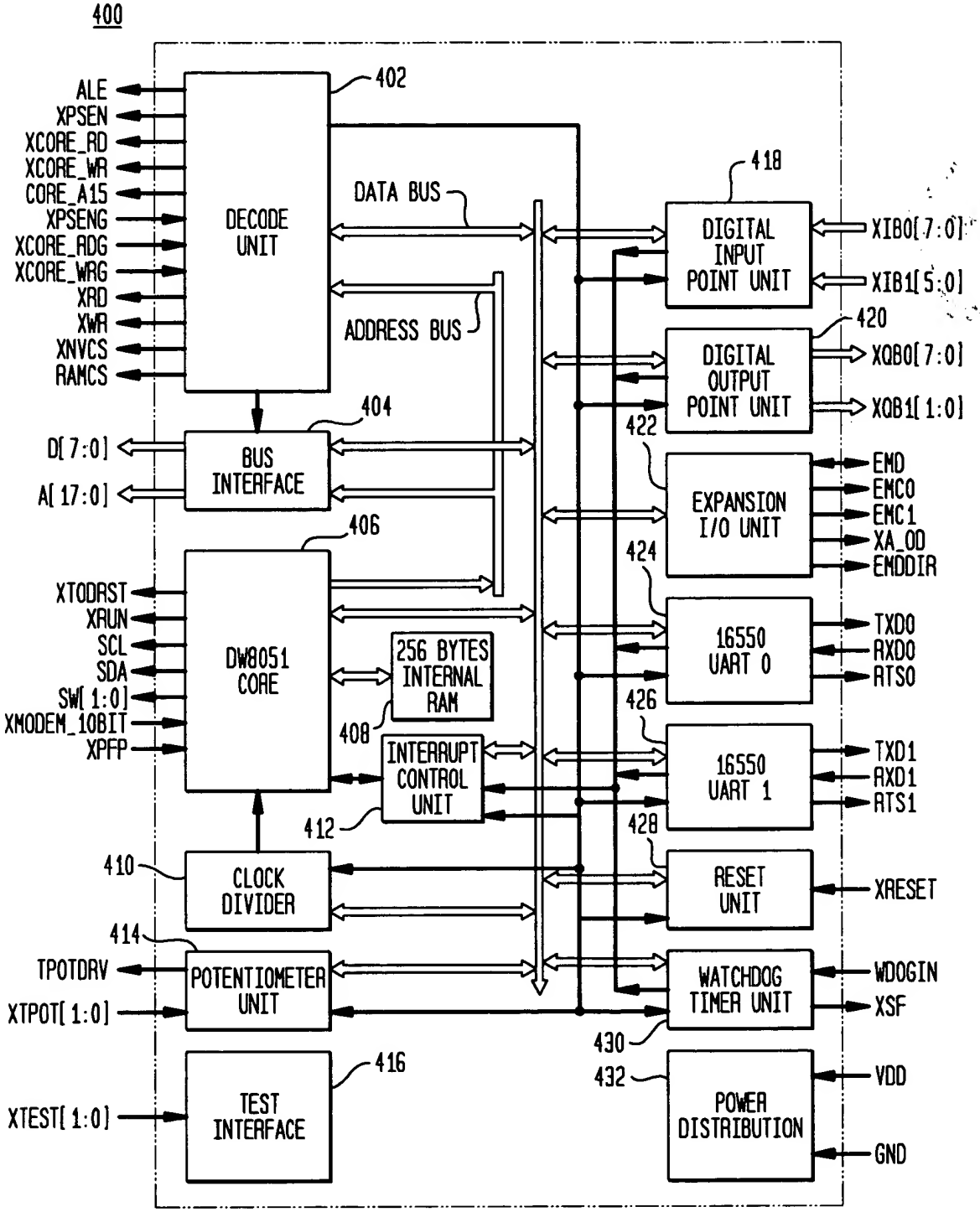

400
402
ALE
XPSEN
XCORE_RD
XCORE_WR
CORE_A15
XPSENG
XCORE_RDG
XCORE_WRG
XRD
XWR
XNVCS
RAMCS
DECODE UNIT
DATA BUS
ADDRESS BUS
404
D[7:0]
A[17:0]
BUS INTERFACE
406
XTODRST
XRUN
SCL
SDA
SW[1:0]
XMODEM_10BIT
XPFP
DW8051 CORE
256 BYTES INTERNAL RAM
408
INTERRUPT CONTROL UNIT
412
410
CLOCK DIVIDER
414
TPOTDRV
XTPOT[1:0]
POTENTIOMETER UNIT
XTEST[1:0]
416
TEST INTERFACE
418
DIGITAL INPUT POINT UNIT
XIB0[7:0]
XIB1[5:0]
420
DIGITAL OUTPUT POINT UNIT
XQB0[7:0]
XQB1[1:0]
422
EXPANSION I/O UNIT
EMD
EMC0
EMC1
XA_OD
EMDDIR
424
16550 UART 0
TXD0
RXD0
RTS0
426
16550 UART 1
TXD1
RXD1
RTS1
428
RESET UNIT
XRESET
WATCHDOG TIMER UNIT
WDOGIN
XSF
430
432
POWER DISTRIBUTION
VDD
GND

FIG. 5

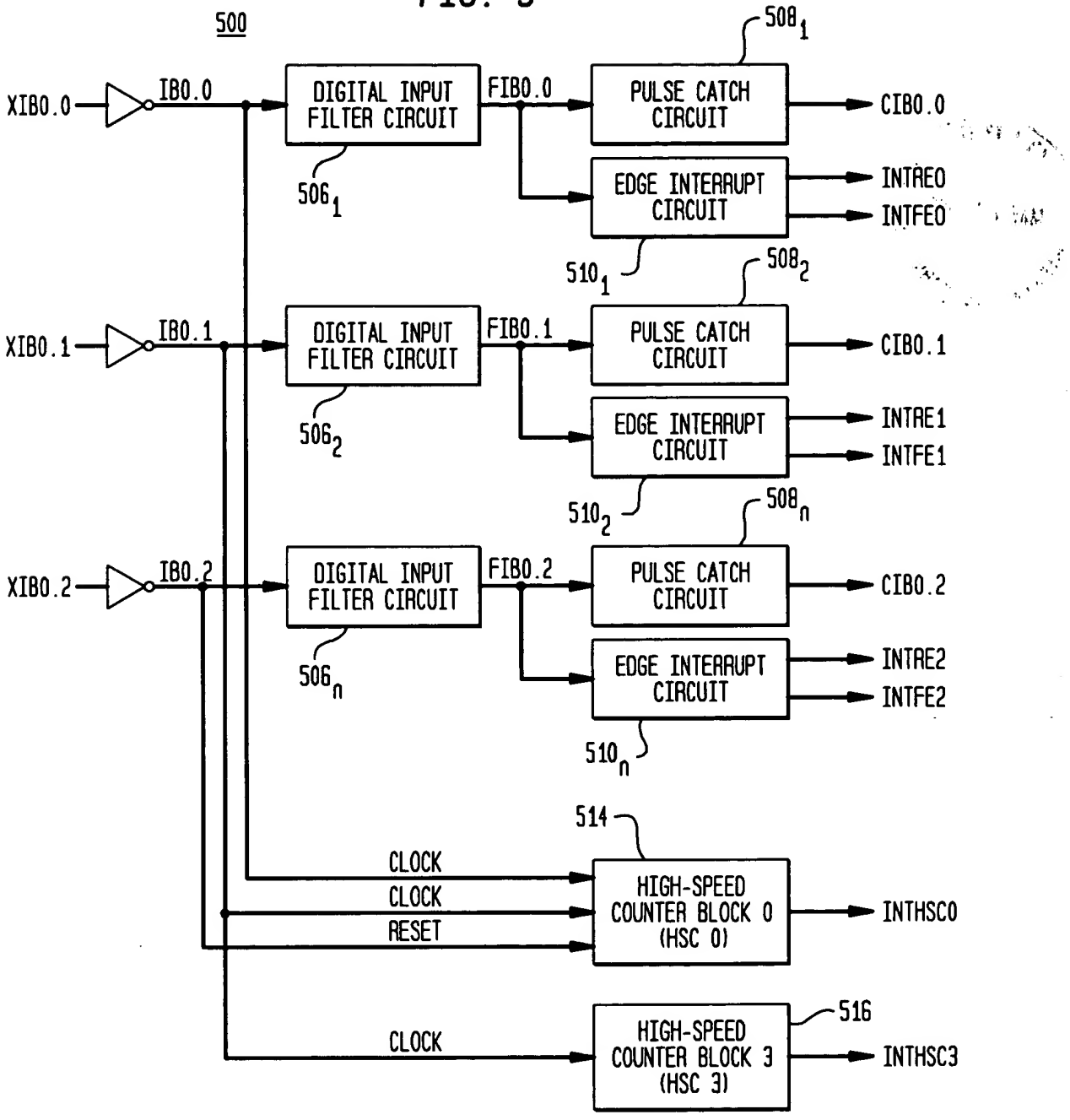

500
XIB0.0
IB0.0
DIGITAL INPUT FILTER CIRCUIT
FIB0.0
PULSE CATCH CIRCUIT
CIB0.0
EDGE INTERRUPT CIRCUIT
INTRE0
INTFE0
XIB0.1
IB0.1
FIB0.1
CIB0.1
INTRE1
INTFE1
XIB0.2
IB0.2
FIB0.2
CIB0.2
INTRE2
INTFE2
CLOCK
CLOCK
RESET
HIGH-SPEED COUNTER BLOCK 0 (HSC 0)
INTHSC0
CLOCK
HIGH-SPEED COUNTER BLOCK 3 (HSC 3)
INTHSC3

FIG. 6A

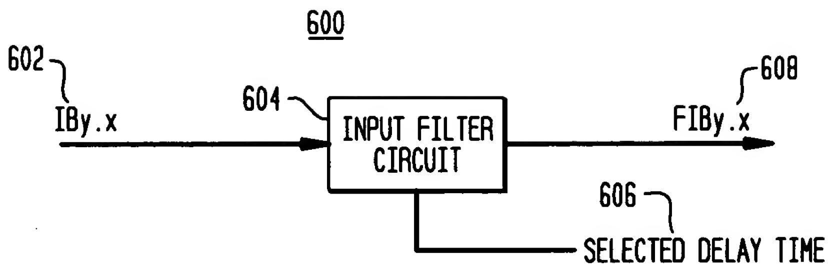


FIG. 6B

DIGITAL FILTER OPERATION DEFINITION

INPUT POINT STATE	CURRENT COUNT	NEXT COUNT	PRESENT OUTPUT VALUE	NEXT OUTPUT VALUE
0 (DECREMENTS COUNTER)	0 n, where 3≥n>0 4 n, where 15≥n>4	0 n - 1 3 n - 1	0 0 1 1	0 0 0 1
1 (INCREMENTS COUNTER)	n, where 11>n≥0 11 n, where 14≥n>11 15	n + 1 12 n + 1 15	0 0 1 1	0 1 1 1

FIG. 6C

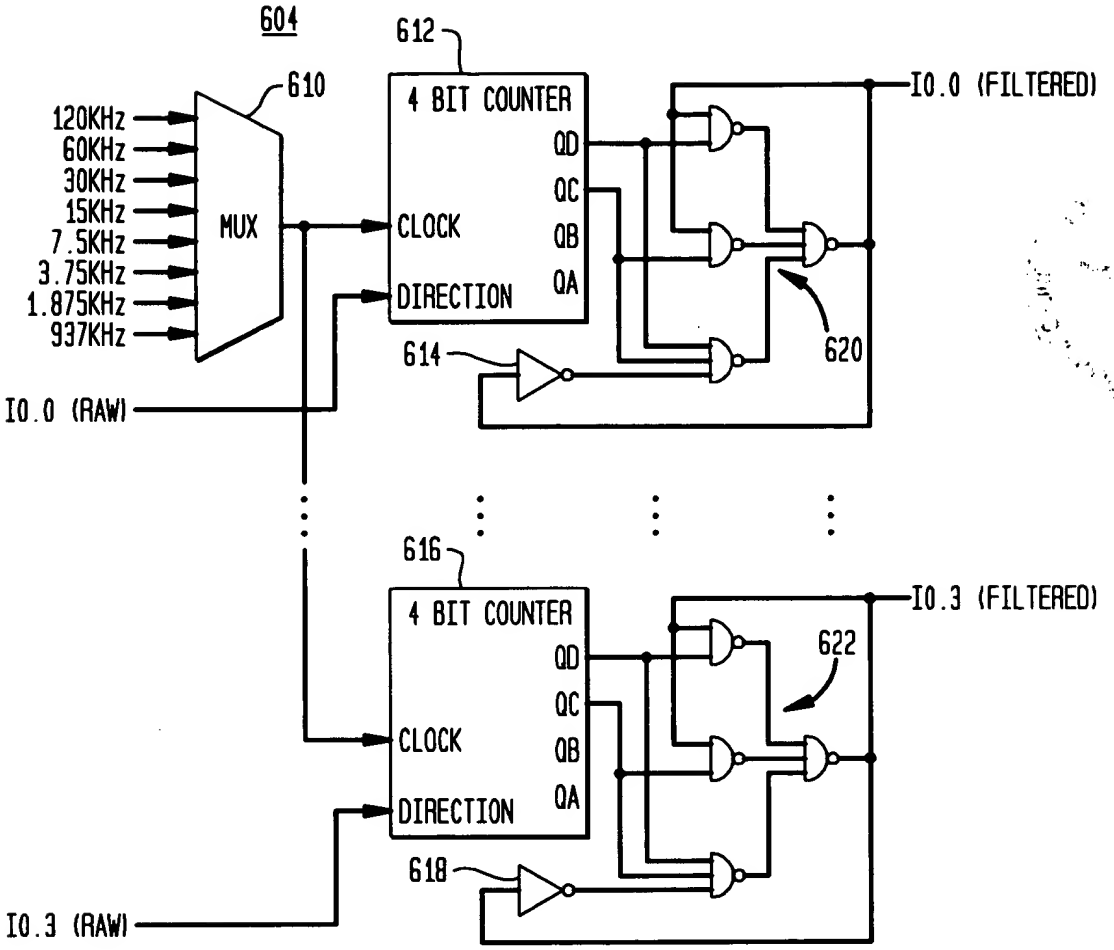


FIG. 6D

FREQUENCY	PERIOD	NUMBER OF COUNTS	DELAY TIME
120 KHz	8.33 μSEC	12	0.1 ms
60 KHz	16.6 μSEC	12	0.2 ms
30 KHz	33.3 μSEC	12	0.4 ms
15 KHz	66.7 μSEC	12	0.8 ms
7.5 KHz	133 μSEC	12	1.6 ms
3.75 KHz	267 μSEC	12	3.2 ms
1.875 KHz	533 μSEC	12	6.4 ms
937 KHz	1067 μSEC	12	12.8 ms

FIG. 7

REGISTER VALUE	CORRESPONDING DELAY TIME
00	0.2 ms
01	0.4 ms
02	0.8 ms
03	1.6 ms[1]
04	1.6 ms[1]
05	3.2 ms
06	6.4 ms
07	12.8 ms
08 TO FF	NO DELAY

FIG. 8

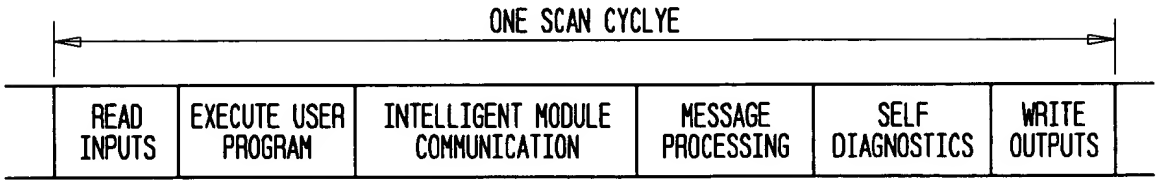


FIG. 9

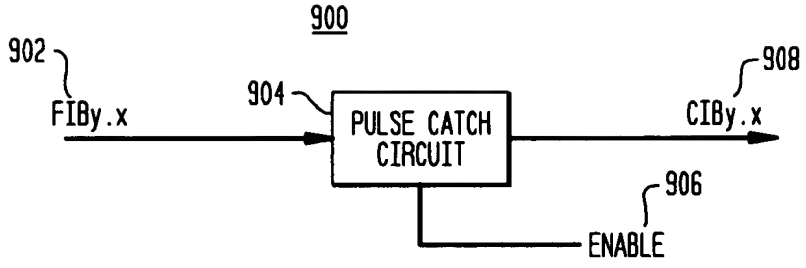

900
902
FIBy.x
904
PULSE CATCH CIRCUIT
908
CIBy.x
906
ENABLE

FIG. 10

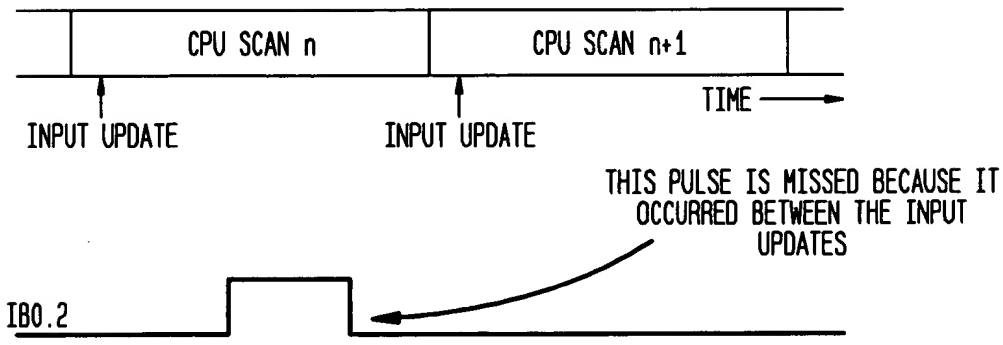

CPU SCAN n
CPU SCAN n+1
TIME
INPUT UPDATE
INPUT UPDATE
THIS PULSE IS MISSED BECAUSE IT OCCURRED BETWEEN THE INPUT UPDATES
IB0.2

FIG. 11

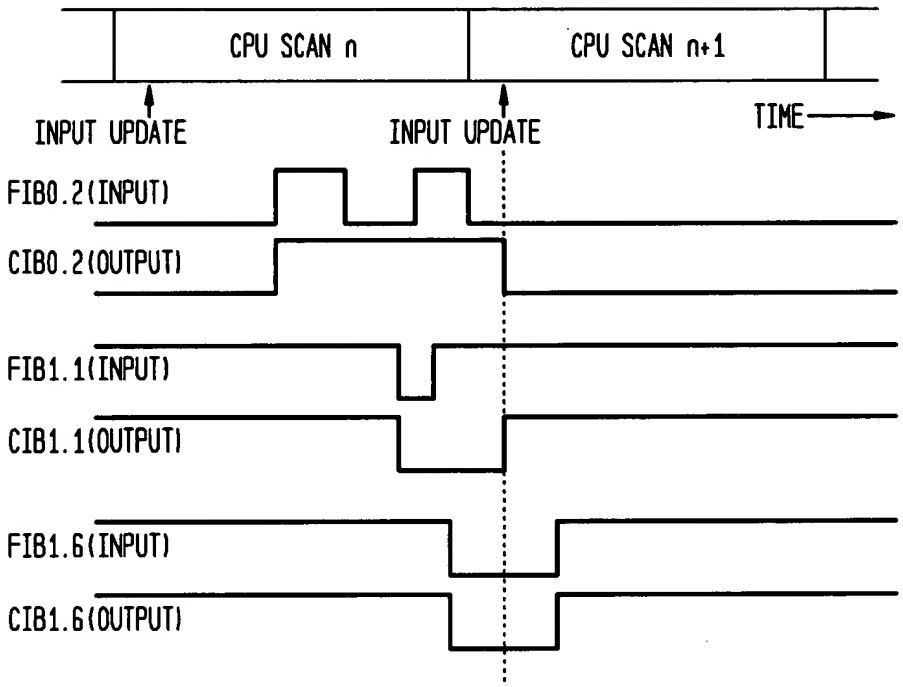


FIG. 12A

PCE	PS CV	PS F	RP	NS CV	NS F	COMMENT
1	I	0	-	CV	0	I = CV AND F IS NOT SET; RP IS A DON'T CARE
1	not I	0	-	I	1	I ≠ CV; CAPTURE NEW VALUE OF I AND SET F = 1
1	-	1	0	CV	1	CV HAS NOT BEEN READ
1	-	1	1	I	0	CV HAS BEEN READ; SET CV = I
0	-	-	-	I	0	PULSE CATCH DISABLED; CV = I

FIG. 12B

PULSE CATCH ENABLE REGISTERS	
ADDRESS	DESCRIPTION
0002H	REGISTER NAME: IB0_Pulse_Catch_Enable_Register (IB0PCE) SIZE: byte (8-bit) ACCESS: read/write RESET VALUE: 00H 7 … 0 EN7 | EN6 | EN5 | EN4 | EN3 | EN2 | EN1 | EN0 ENx: 1 = ENABLES PULSE CATCH OPERATION ON INPUT POINT IB0.x 0 = DISABLES PULSE CATCH OPERATION ON INPUT POINT IB0.x
0003H	REGISTER NAME: IB1_Pulse_Catch_Enable_Register (IB1PCE) SIZE: byte (8-bit) ACCESS: read/write RESET VALUE: 00H 7 … 0 x | x | EN5 | EN4 | EN3 | EN2 | EN1 | EN0 ENx: 1 = ENABLES PULSE CATCH OPERATION ON INPUT POINT IB1.x 0 = DISABLES PULSE CATCH OPERATION ON INPUT POINT IB1.x

FIG. 12C

REGISTER IB0PS:	READ OF THIS REGISTER RETURNS CIB0[7:0] AND RETRIGGERS PULSE CATCH CIRCUITS FOR IB0 INPUT POINTS	USED BY SW FOR INPUT UPDATE
REGISTER IB1PS:	READ OF THIS REGISTER RETURNS CIB1[5:0] AND RETRIGGERS PULSE CATCH CIRCUITS FOR IB1 INPUT POINTS	USED BY SW FOR INPUT UPDATE
REGISTER IB0PSNR:	READ OF THIS REGISTER RETURNS CIB0[7:0] AND LEAVES PULSE CATCH CIRCUITS UNAFFECTED	USED BY SW FOR IMMEDIATE ACCESS
REGISTER IB1PSNR:	READ OF THIS REGISTER RETURNS CIB1[5:0] AND LEAVES PULSE CATCH CIRCUITS UNAFFECTED	USED BY SW FOR IMMEDIATE ACCESS

FIG. 12D

INPUT POINT STATUS REGISTERS	
ADDRESS	DESCRIPTION
0004H	REGISTER NAME: IB0_Input_Point_Status_Register (IB0PS) SIZE: byte (8-bit) ACCESS: read only RESET VALUE: 00H 7 … 0 CI7 | CI6 | CI5 | CI4 | CI3 | CI2 | CI1 | CI0 CIx: CONTAINS CONDITIONED INPUT POINT STATE CIB0.x
0005H	REGISTER NAME: IB1_Input_Point_Status_Register (IB1PS) SIZE: byte (8-bit) ACCESS: read only RESET VALUE: 00H 7 … 0 x | x | CI5 | CI4 | CI3 | CI2 | CI1 | CI0 CIx: CONTAINS CONDITIONED INPUT POINT STATE CIB1.x.
0006H	REGISTER NAME: IB0_Input_Point_Status_Register_No_Retrigger (IB0PSNR) SIZE: byte (8-bit) ACCESS: read only RESET VALUE: 00H 7 … 0 CI7 | CI6 | CI5 | CI4 | CI3 | CI2 | CI1 | CI0 CIx: CONTAINS CONDITIONED INPUT POINT STATE CIB1.x
0007H	REGISTER NAME: IB1_Input_Point_Status_Register_No_Retrigger (IB1PSNR) SIZE: byte (8-bit) ACCESS: read only RESET VALUE: 00H 7 … 0 x | x | CI5 | CI4 | CI3 | CI2 | CI1 | CI0 CIx: CONTAINS CONDITIONED INPUT POINT STATE CIB0.x.

FIG. 12E

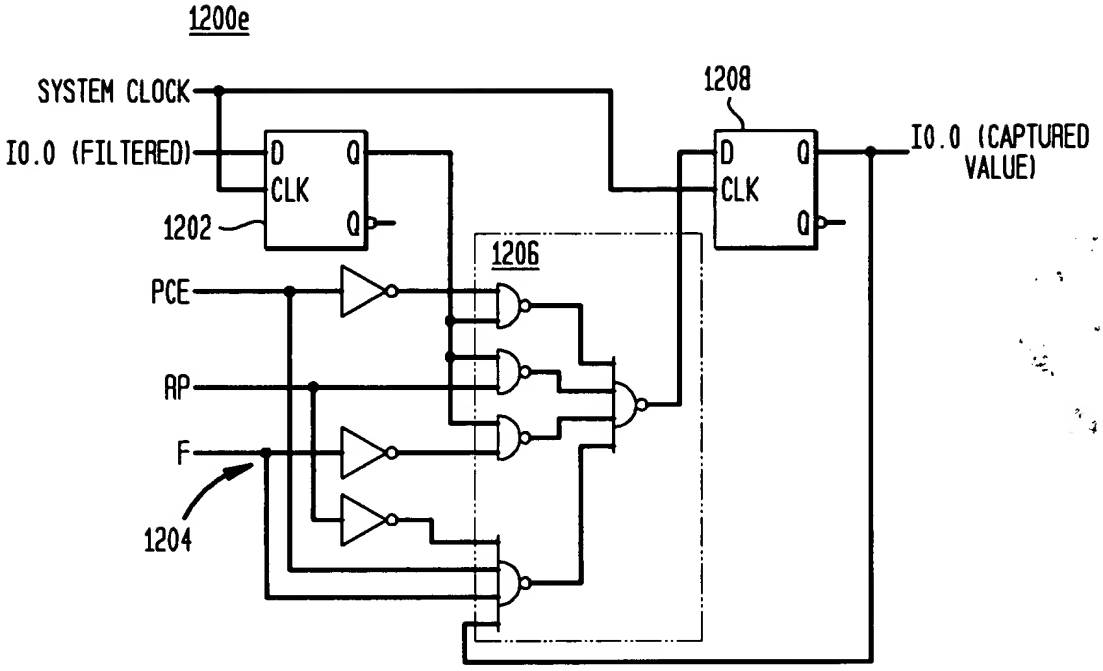


FIG. 12F

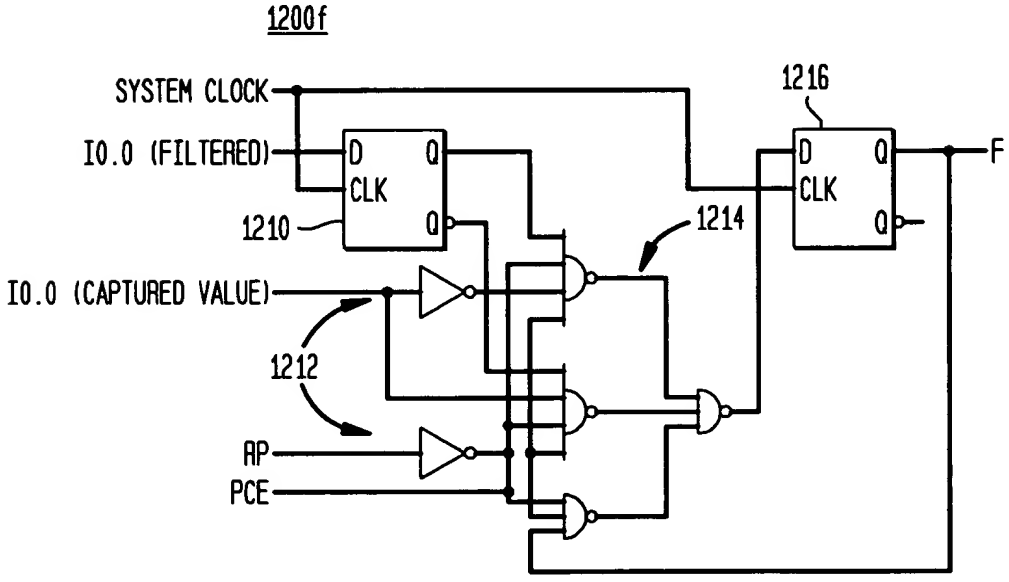


FIG. 13

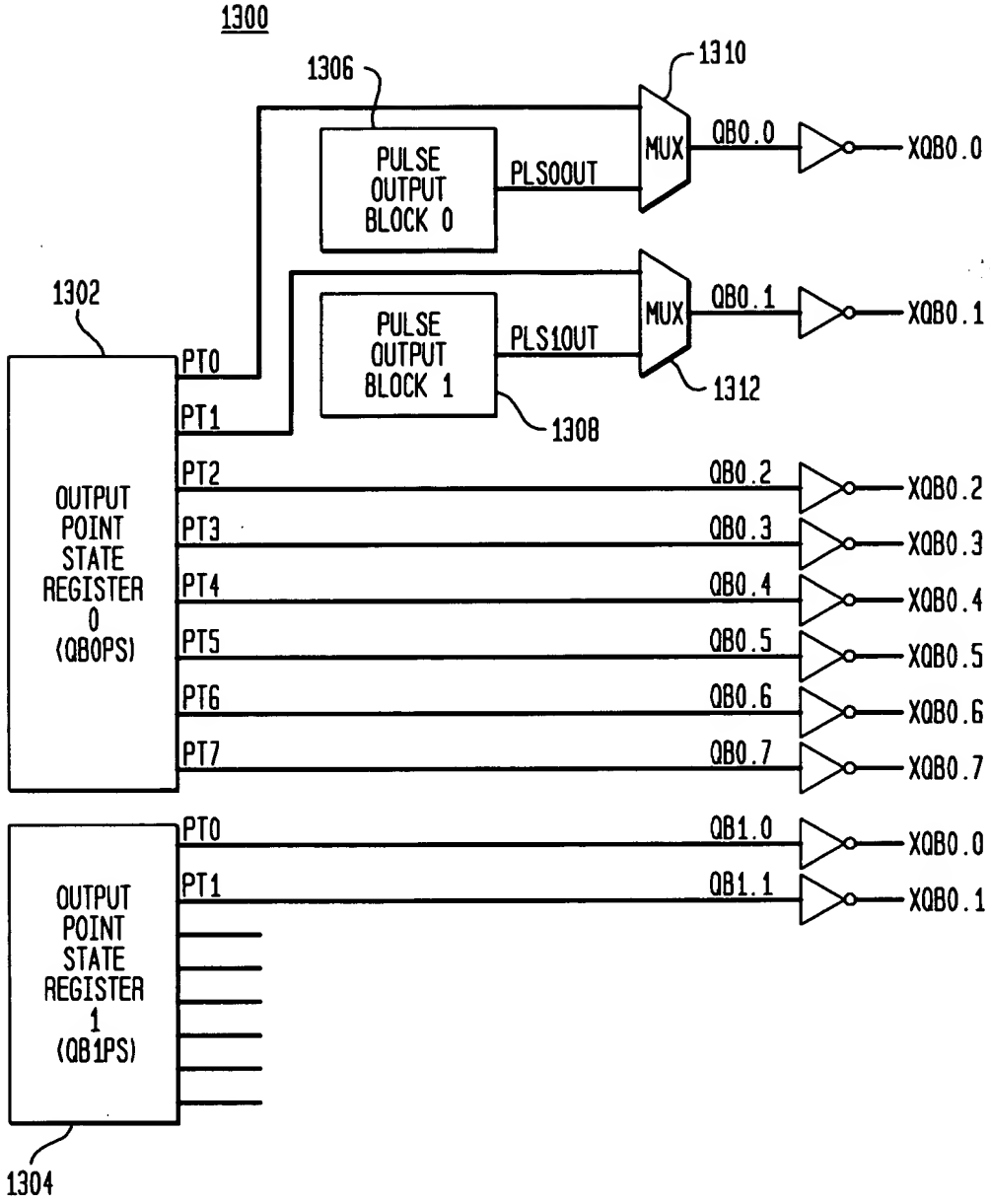

1300
1306
1310
PULSE OUTPUT BLOCK 0
PLS0OUT
MUX
QB0.0
XQB0.0
1302
PULSE OUTPUT BLOCK 1
PLS1OUT
MUX
QB0.1
XQB0.1
1312
1308
PT0
PT1
OUTPUT POINT STATE REGISTER 0 (QB0PS)
PT2
PT3
PT4
PT5
PT6
PT7
QB0.2
QB0.3
QB0.4
QB0.5
QB0.6
QB0.7
XQB0.2
XQB0.3
XQB0.4
XQB0.5
XQB0.6
XQB0.7
PT0
PT1
OUTPUT POINT STATE REGISTER 1 (QB1PS)
QB1.0
QB1.1
XQB0.0
XQB0.1
1304

FIG. 14A

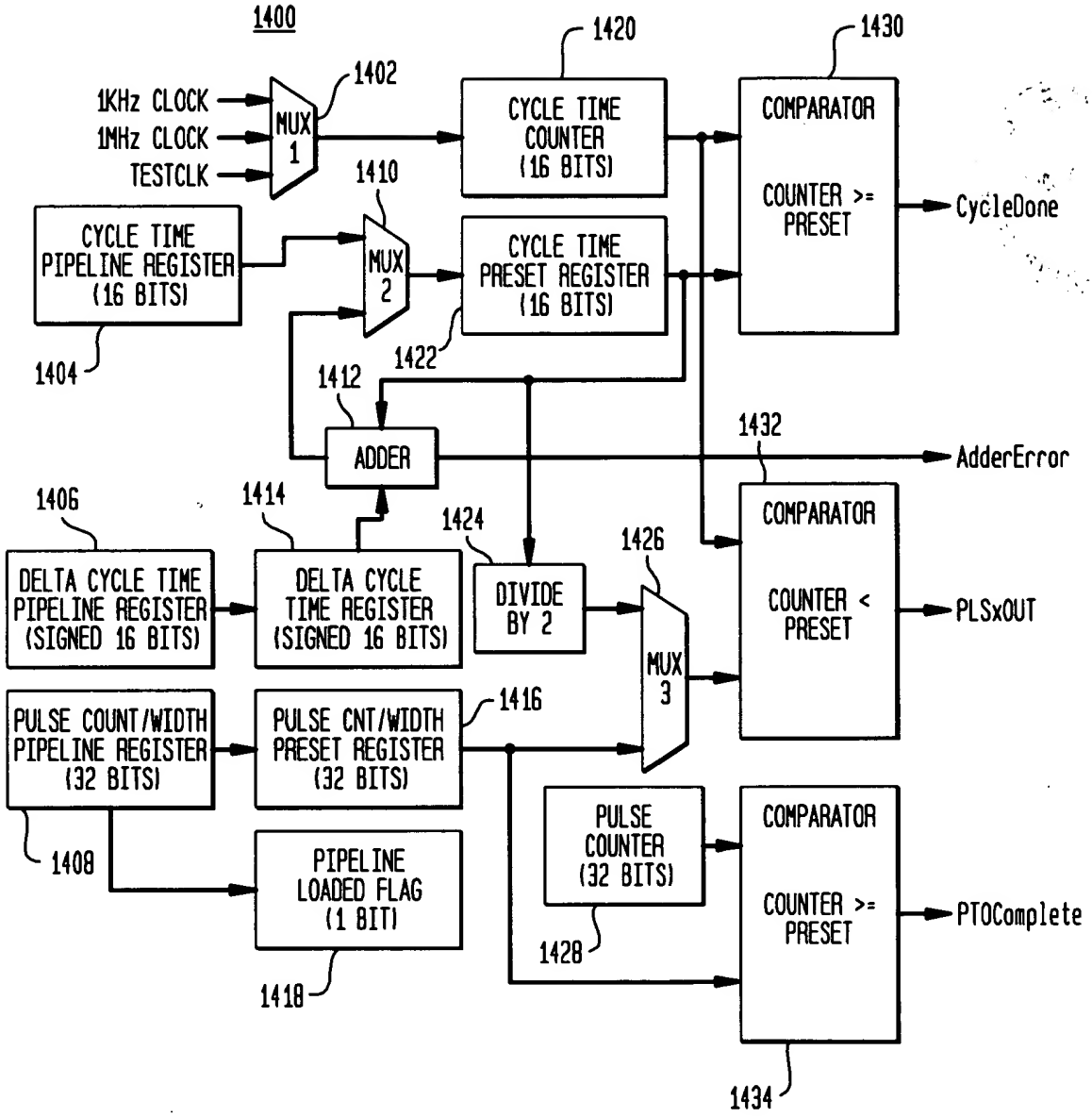

1400
1420
1430
1KHz CLOCK
1MHz CLOCK
TESTCLK
1402
MUX 1
CYCLE TIME COUNTER (16 BITS)
COMPARATOR
COUNTER >= PRESET
CycleDone
1410
CYCLE TIME PIPELINE REGISTER (16 BITS)
MUX 2
CYCLE TIME PRESET REGISTER (16 BITS)
1404
1422
1412
ADDER
1432
AdderError
1406
1414
1424
1426
DELTA CYCLE TIME PIPELINE REGISTER (SIGNED 16 BITS)
DELTA CYCLE TIME REGISTER (SIGNED 16 BITS)
DIVIDE BY 2
MUX 3
COMPARATOR
COUNTER < PRESET
PLSxOUT
PULSE COUNT/WIDTH PIPELINE REGISTER (32 BITS)
PULSE CNT/WIDTH PRESET REGISTER (32 BITS)
1416
1408
PIPELINE LOADED FLAG (1 BIT)
PULSE COUNTER (32 BITS)
COMPARATOR
COUNTER >= PRESET
PTOComplete
1428
1418
1434

FIG. 14B

REGISTER NAME	VALID VALUE RANGE
CYCLE TIME PRESET REGISTER CYCLE TIME PIPELINE REGISTER	2 TO 65535
DELTA CYCLE TIME REGISTER DELTA CYCLE PIPELINE REGISTER	-32768 TO 32767
PULSE COUNT/WIDTH PRESET REGISTER PULSE COUNT/WIDTH PIPELINE REGISTER	1 TO $(2^{32}-1)$ 0 TO 65535 PTO MODE PWM MODE

FIG. 15

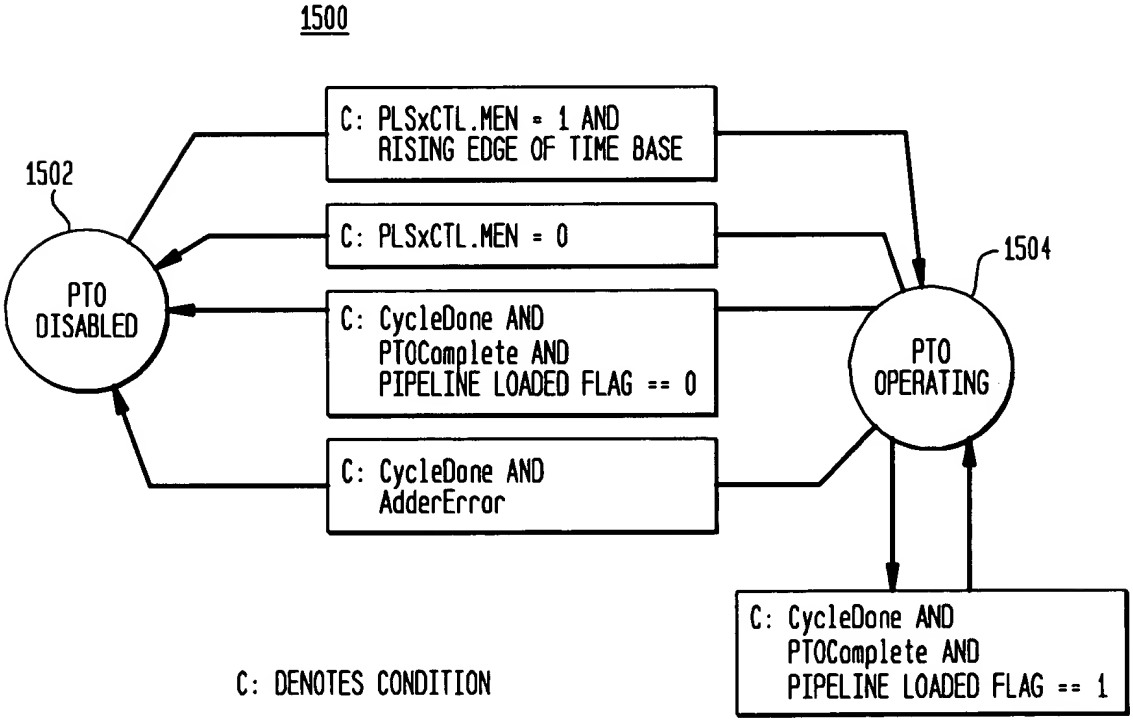


FIG. 16

```
cycle time counter = cycle time counter + 1;
IF (cycle time counter >= cycle time preset) THEN
BEGIN  //this is the CycleDone event
       --------
       pulse counter = pulse counter + 1;
       IF (pulse counter >= pulse count preset) THEN
       BEGIN  //this is the PTOComplete event
              --------
              assert INTxPLS signal, if PTOComplete interrupts are enabled;
              IF (pipeline loaded flag is set) THEN
              BEGIN
                     --------
                     transfer values from pipline registers into operating registers;
                     set pulse counter = 0;
                     clear pipeline loaded flag;
                     --------
              END
              ELSE  //pipeline loaded flag is not set
              BEGIN
                  --------
                  GOTO PTO Disabled state; //disable the PLS block now
                  --------
              END
              ENDIF
              --------
       END

       ELSE   //not yet at PTOComplete
       BEGIN
              --------
              cycle time preset = cycle time preset + delta cycle time;
              IF (cycle time preset exceeds bounds) THEN
              BEGIN  //this is the AdderError event
                     --------
                     assert INTxPLS signal, if AdderError interrupts are enabled;
                     GOTO PTO Disabled state;  //disable the PLS block now
                     --------
              END
              ENDIF
              --------
       END
       ENDIF
       set cycle time counter = 0;
       --------
END
ENDIF

IF (cycle time counter >=(1/2 * cycle time preset)) THEN
        PLSxOUT = 0;
ELSE    //output still in logic high portion of the current cycle
        PLSxOUT = 1;
ENDIF
```

FIG. 17

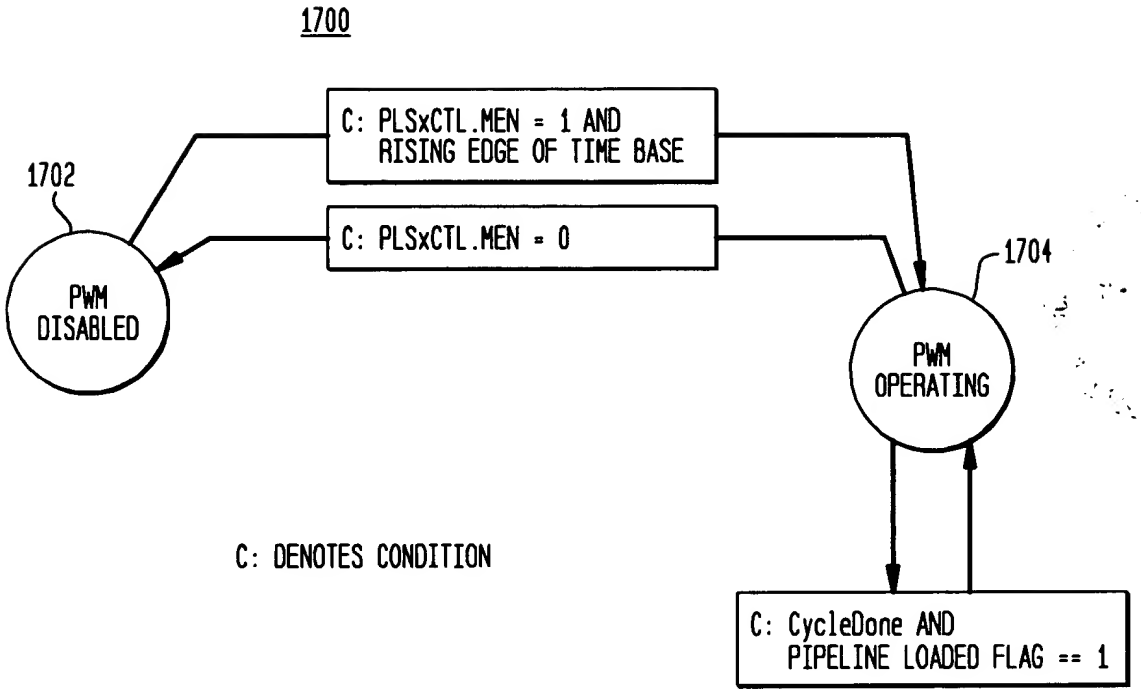


FIG. 18

```
cycle time counter = cycle time counter + 1;
IF (cycle time counter >= cycle time preset) THEN
BEGIN  //this is the CycleDone event
       --------
       IF (pipeline loaded flag is set) THEN
       BEGIN
              --------
              transfer values from pipeline registers into operating registers;
              clear pipeline loaded flag;
              --------
       END
       ENDIF
       set cycle time counter = 0;
       --------
END
ENDIF

IF (cycle time counter >= (pulse width preset)) THEN
       PLSxOUT = 0;
ELSE   //output still in logic high portion of the current cycle
       PLSxOUT = 1;
ENDIF
```

FIG. 19

HIGH SPEED OPERATIONS				
INSTRUCTION	MNEMONIC & OPERAND(S)	DESCRIPTION	STL STATUS ELEMENT	VALID OPERANDS
PULSE TRAIN OUTPUT PROFILE	PTOP t, n	WHEN SO = 1, THE PTO PROFILE SPECIFIED IN THE TABLE FOR OUTPUT n IS EXECUTED. ENO <- SO * /e	STK, <CURRENT STEP>	ENABLE: SO TABLE: VB, IB, QB, MB, (UI) SMB, SB, LB, *VD, *AC OUTPUT: KW (UI) 0 - 1

DEFINITION OF THE TABLE FOR PTOP:

BYTE OFFSET	SEGMENT	DESCRIPTION OF TABLE ENTRIES
0		NUMBER OF PROFILE SEGMENTS (40 SEGMENTS MAXIMUM)
1		CURRENT STEP NUMBER BEING EXECUTED
2	#1	NUMBER OF STEPS (4 STEPS MINIMUM)
4		STARTING CYCLE TIME FOR THIS SEGMENT (2 TO 65535 µSEC)
6		CHANGE IN CYCLE TIME PER STEP (SIGNED VALUE) (0 TO 65535 µSEC)
8	#2	NUMBER OF STEPS (4 STEPS MINUMUM)
10		STARTING CYCLE TIME FOR THIS SEGMENT (2 TO 65535 µSEC)
12		CHANGE IN CYCLE TIME PER STEP (SIGNED VALUE) (0 TO 65535 µSEC)
. . .	. . .	. . .

FIG. 20

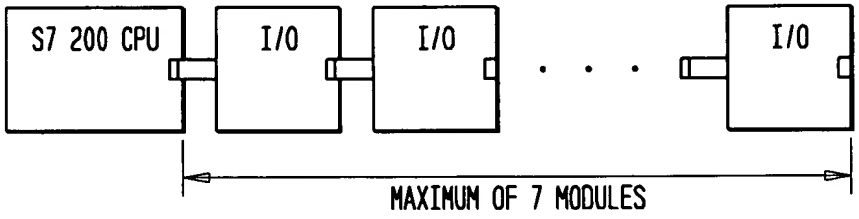


FIG. 21

SIGNAL NAME	USE
EMD	EXPANSION MODULE DATA - A BI-DIRECTIONAL SIGNAL USED TO COMMUNICATE THE ADDRESS AND DATA TO AND FROM THE MODULE.
EMC[1:0]	EXPANSION MODULE CLOCKS - ONE CLOCK IS USED TO ACCESS EXPANSION I/O EXTERNAL TO THE PLC, WHILE THE OTHER IS USED TO ACCESS I/O THAT IS LOCAL TO THE PLC THAT DOES NOT CONNECT DIRECTLY TO THE ASIC.
XA_OD	ADDRESS/OUTPUT DISABLE - A DUAL FUNCTION SIGNAL USED TO RESET THE STATE MACHINE IN THE MODULES ON THE FIRST CLOCK OF EACH ACCESS CYCLE (ACTIVE LOW) AND USED TO INDICATE OUTPUT DISABLE WHEN A FATAL ERROR HAS BEEN DETECTED (ACTIVE LOW FOR AN RC TIME CONSTANT).
EMDDIR	EXPANSION MODULE DATA DIRECTION - THIS SIGNAL INDICATES THE DIRECTION OF DATA FLOW ON THE EMD SIGNAL LINE. 0 - DATA IS DRIVEN BY THE MODULE TO THE PLC 1 - DATA IS DRIVEN BY THE PLC TO THE MODULE
EMA[2:0]	ESPANSION MODULE ADDRESS - THESE SIGNALS ARE DAISY CHAINED FROM PLC TO MODULE TO MODULE. THE VALUE INPUT TO A MOCULE BECOMES THAT MODULE'S ADDRESS. THE MODULE WILL OUTPUT ITS ADDRESS PLUS ONE TO THE NEXT MODULE. THE PLC DRIVES THESE SIGNALS TO 0 SO THAT THE MODULE CONNECTED TO THE PLC HAS THE ADDRESS OF 0.
+5V	5 VOLT POWER SUPPLY - TWO SIGNALS CARRY +5 VOLTS.
GND	POWER SUPPLY RETURN - TWO SIGNALS CARRY GROUND.

FIG. 22

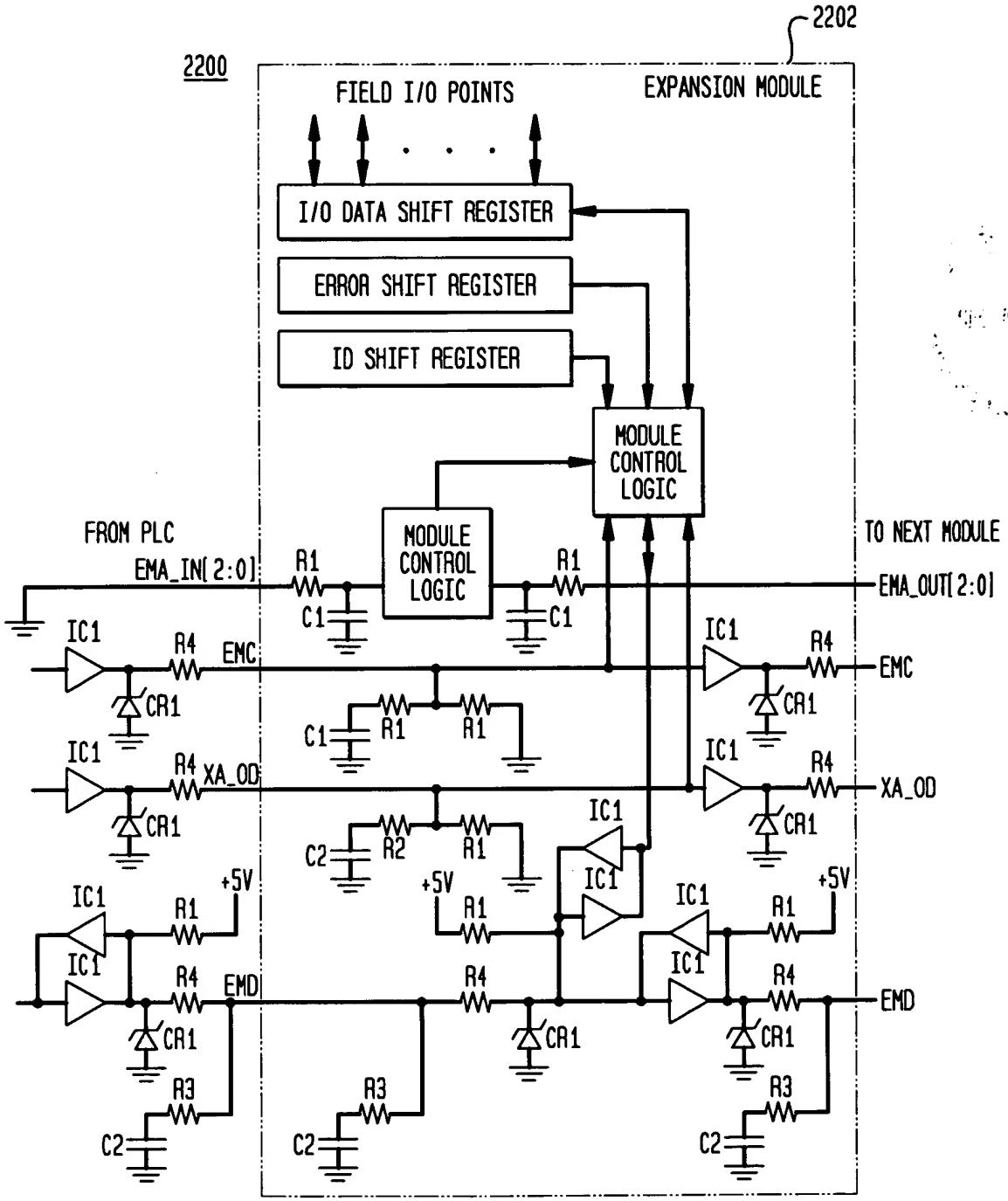

2202
2200
FIELD I/O POINTS
EXPANSION MODULE
I/O DATA SHIFT REGISTER
ERROR SHIFT REGISTER
ID SHIFT REGISTER
MODULE CONTROL LOGIC
MODULE CONTROL LOGIC
FROM PLC
EMA_IN[2:0]
TO NEXT MODULE
EMA_OUT[2:0]
IC1
R1
C1
R4
EMC
CR1
R2
XA_OD
+5V
R3
EMD

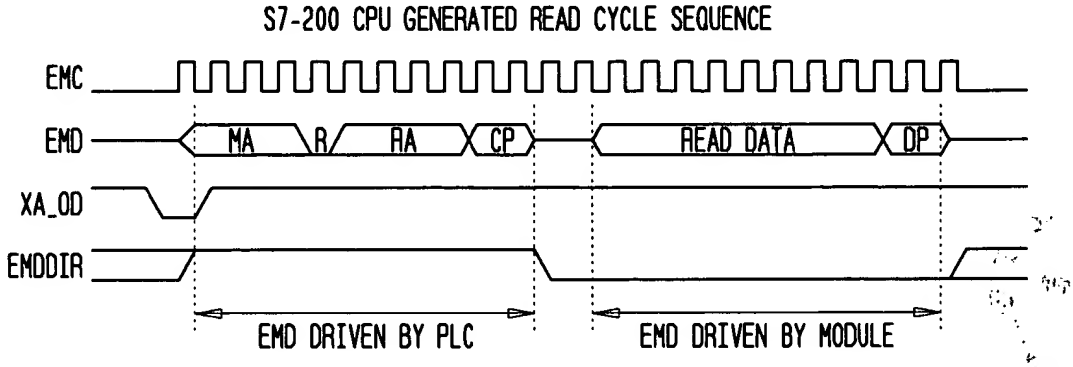
FIG. 23

REFERENCE DESIGNATOR	COMPONENT VALUE/TYPE	COMPONENT DESCRIPTION
IC1	74ABT125/74ABT126	TRI-STATE BUFFER
CR1	TVS 5.6V ZENER	DIODE
R1	4.7K ohm	RESISTOR
R2	110 ohm	RESISTOR
R3	220 ohm	RESISTOR
R4	22 ohm	RESISTOR
C1	0.1 µF	CAPACITOR
C2	100pF	CAPACITOR

FIG. 24

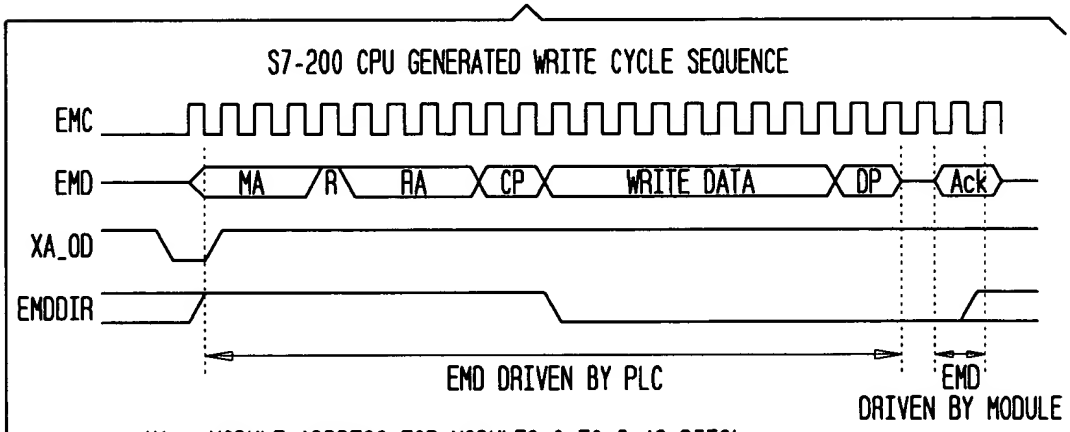
SIGNAL NAME	DRIVING DEVICE	DRIVE LEVELS				RECEIVE LEVELS	
		V_{OL} (VOLTS) (MAX)	I_{OL} (mA) (MIN)	V_{OH} (VOLTS) (MAX)	I_{OH} (mA) (MIN)	V_{IL} (VOLTS) (MAX)	V_{IH} (VOLTS) (MIN)
EMA[2:0]	MODULE	0.5	3.0	2.4	-3.0	0.8	2.0
XA_OD	CPU	0.55	64.0	2.0	-32.0	0.8	2.0
EMC	CPU	0.55	64.0	2.0	-32.0	0.8	2.0
EMD	CPU	0.55	64.0	2.0	-32.0	0.8	2.0
	MODULE	0.55	64.0	2.0	-32.0	0.8	2.0

FIG. 25



MA - MODULE ADDRESS FOR MODULES 0 TO 6 (3 BITS)
R - READ/WRITE BIT (READ ACTIVE LOW)
RA - REGISTER ADDRESS FOR REGISTERS R0 TO R15 (4 BITS)
CP - CONTROL PARITY GENERATED BY THE CPU ON MA, R, AND RA (2 BITS)
READ DATA - DATA READ FROM THE MODULE (8 BITS)
DP - DATA PARITY GENERATED BY THE MODULE ON READ DATA (2 BITS)

FIG. 26



MA - MODULE ADDRESS FOR MODULES 0 TO 6 (3 BITS)
R - READ/WRITE BIT (READ ACTIVE LOW)
RA - REGISTER ADDRESS FOR REGISTERS R0 TO R15 (4 BITS)
CP - CONTROL PARITY GENERATED BY THE CPU ON MA, R, AND RA (2 BITS)
WRITE DATA - DATA WRITTEN TO THE MODULE (8 BITS)
DP - DATA PARITY GENERATED BY THE CPU ON WRITE DATA (2 BITS)
Ack - MODULE ACKNOWLEDGE OF A GOOD WRITE CYCLE (2 BITS)

FIG. 27

	a	b	c	d	e
P0	BIT 0	BIT 2	BIT 4	BIT 6	BIT 7
P1	BIT 1	BIT 2	BIT 3	BIT 5	BIT 7
5 BIT ODD PARITY = !(((a xor b) xor (c xor d) xor e					

FIG. 28

PLC TYPE	PORT 0	PORT 1
CPU 212	256	-
CPU 214	256	-
CPU 215/216	256	256

FIG. 29

INTERRUPTS				
INSTRUCTION	MNEMONIC & OPERAND(S)	DESCRIPTION	STL STATUS ELEMENT	VALID OPERANDS
PASS TOKEN	PASS	USER PROGRAM HAS COMPLETED ITS USE OF THE TOKEN HOLD PERIOD AND IS RETURNING CONTROL TO THE SYSTEM.	STK	

FIG. 30A

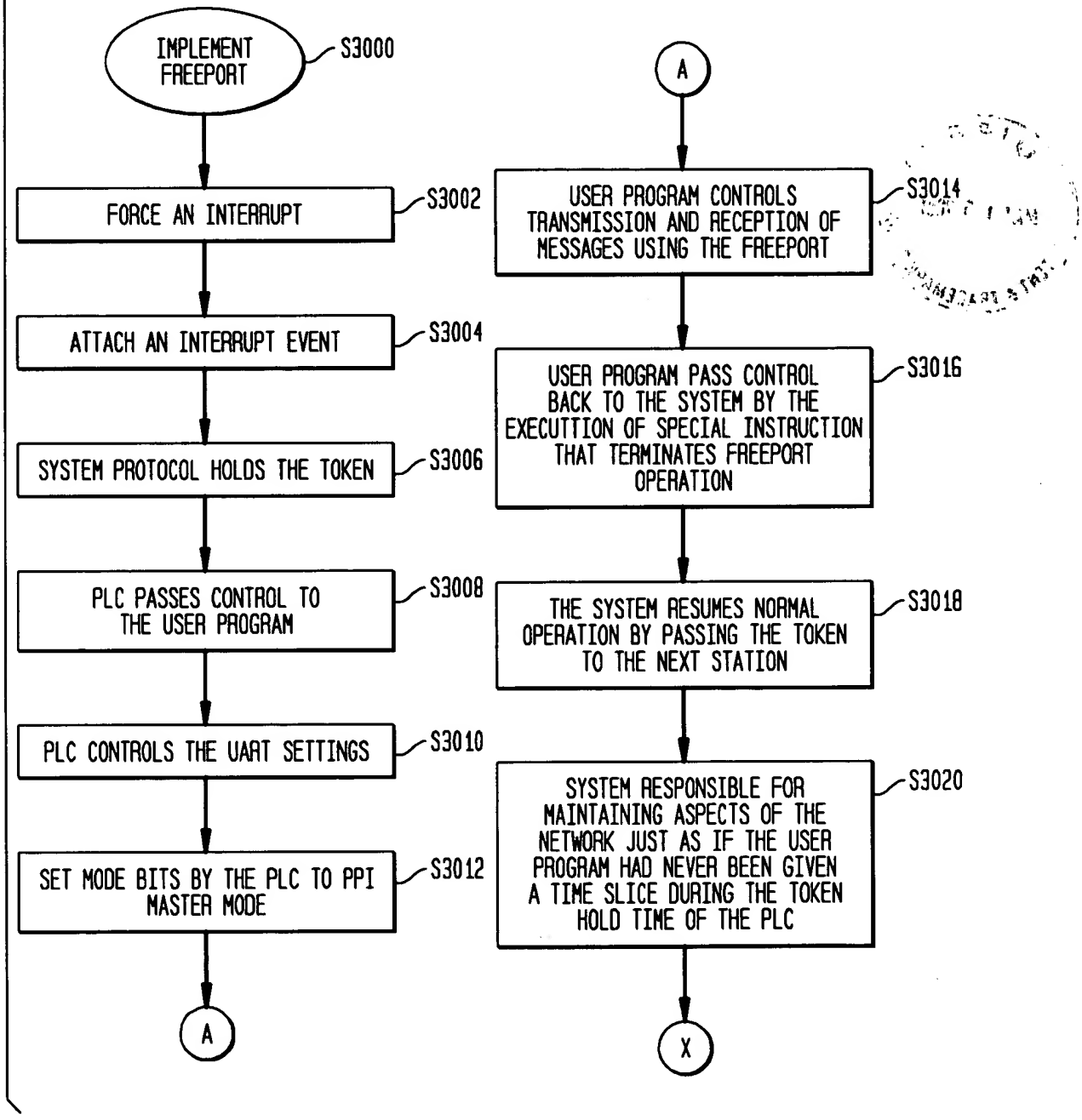


FIG. 30B

SM BIT DEFINITION (READ/WRITE)	
SM BITS	DESCRIPTION
SM30	PORT 0: COMMUNICATION PORT USAGE MSB 7 … LSB 0 SMB30: p | p | d | r | r | r | m | m pp (PARITY) '00' - NO PARITY '01' - EVEN PARITY '10' - NO PARITY '11' - ODD PARITY d (DATA BITS/CHAR) '0' - 8 BITS/CHAR '1' - 7 BITS/CHAR rrr (BAUD RATE) '000' - 38,400 '001' - 19,200 '010' - 9600 '011' - 4800 '100' - 2400 '101' - 1200 '110' - 600 '111' - 300 mm (PROTOCOL) '00' - PPI SLAVE (DEFAULT) '01' - FREEPORT '10' - PPI MASTER '11' - RESERVED (PPI SLAVE) WHEN THE USER SELECTS THE CODE mm = 10 (PPI MASTER), THE PLC WILL BECOME A MASTER ON THE NETWORK ALLOWING THE NETR AND NETW INSTRUCTIONS TO BE EXECUTED. BITS 2 THROUGH 7 ARE IGNORED IN PPI MODES. IN PPI MASTER MODE WITH THE TOKEN ACQUIRED INTERRUPT ENABLED, THESE BITS ARE USED TO SETUP THE UART PROIR TO TRANSFERRING CONTROL TO THE USER'S PROGRAM.
SM130	PORT 0: COMMUNICATION PORT USAGE (CPU 216 ONLY) MSB 7 … LSB 0 SMB130: p | p | d | r | r | r | m | m pp (PARITY) '00' - NO PARITY '01' - EVEN PARITY '10' - NO PARITY '11' - ODD PARITY d (DATA BITS/CHAR) '0' - 8 BITS/CHAR '1' - 7 BITS/CHAR rrr (BAUD RATE) '000' - 38,400 '001' - 19,200 '010' - 9600 '011' - 4800 '100' - 2400 '101' - 1200 '110' - 600 '111' - 300 mm (PROTOCOL) '00' - PPI SLAVE (DEFAULT) '01' - FREEPORT '10' - PPI MASTER '11' - RESERVED (PPI SLAVE) WHEN THE USER SELECTS THE CODE mm = 10 (PPI MASTER), THE PLC WILL BECOME A MASTER ON THE NETWORK ALLOWING THE NETR AND NETW INSTRUCTIONS TO BE EXECUTED. BITS 2 THROUGH 7 ARE IGNORED IN PPI MODES. IN PPI MASTER MODE WITH THE TOKEN ACQUIRED INTERRUPT ENABLED, THESE BITS ARE USED TO SETUP THE UART PROIR TO TRANSFERRING CONTROL TO THE USER'S PROGRAM.

FIG. 30C

SM BIT DEFINITION (READ/WRITE) (CONTINUED)	
SM BITS	DESCRIPTION
SM87	PORT 0: RECEIVE MESSAGE CONTROL BYTE, (CPU 212/214/216) MSB 7 LSB 0 en | sc | ec | il | c/m | tmr | bk | 0 en: ENABLE/DISABLE RECEIVE MESSAGE BIT IS CHECKED EACH TIME THE RCV INSTRUCTION IS EXECUTED. IF THIS BIT IS A "0", THEN THE RECEIVE MESSAGE FUNCTION IS DISABLED. IF THIS BIT IS A "1", THEN THE RECEIVE MESSAGE FUNCTION IS ENABLED. sc: 0 - IGNORE SMB88; 1 - USE THE VALUE OF SMB88 TO DETECT START OF MESSAGE ec: 0 - IGNORE SMB89; 1 - USE THE VALUE OF SMB89 TO DETECT END OF MESSAGE il: 0 - IGNORE SMW90; 1 - USE THE VALUE OF SMW90 TO DETECT AN IDLE LINE CONDITION[1] c/m: 0 - USE TIMER AS AN INTER-CHARACTER TIMER; 1 - USE TIMER AS A MESSAGE TIMER tmr: 0 - IGNORE SMW92; 1 - TERMINATE RECEIVE IF THE TIME PREIOD IN SMW92 IS EXCEEDED bk: 0 - IGNORE BREAK CONDITIONS; 1 - USE BREAK CONDITION AS START OF MESSAGE DETECTION [1]BY SETTING THE sc AND bk BITS TO 0 AND THE en, il, c/m AND tmr BITS TO 1 WITH AN IDLE LINE TIMER VALUE OF ZERO, SMW92 WILL BE USED TO TIME OUT THE RCV INSTRUCTION WITHOUT RECEIVING ANY CHARACTERS. IF THE TIMER IS NOT USED (tmr = 0), THEN ANY CHARACTER RECEIVED WILL BE USED AS START OF MESSAGE. THE BITS OF THE MESSAGE INTERRUPT CONTROL BYTE ARE USED TO DEFINE THE CRITERIA BY WHICH THE MESSAGE IS IDENTIFIED. BOTH START OF MESSAGE AND END OF MESSAGE CRITERIA ARE DEFINED. TO DETERMINE THE START OF A MESSAGE EITHER OF TWO SETS OF LOGICALLY ANDED START OF MESSAGE CRITERIA MUST BE TRUE AND MUST OCCUR IN SEQUENCE (IDLE LINE FOLLOWED BY START CHARACTER OR BREAK FOLLOWED BY START CHARACTER). TO DETERMINE THE END OF A MESSAGE THE ENABLED END OF MESSAGE CRITERIA ARE LOGICALLY Ored. THE EQUATIONS FOR START AND STOP CRITERIA ARE GIVEN BELOW: Start of Message = il * sc + bk * sc End of Message = ec + tmr + maximum character count reached NOTE: RECEIVE WILL AUTOMATICALLY BE TERMINATED BY AN OVERRUN OR A PARITY ERROR (IF ENABLED).
SM88	PORT 0: START OF MESSAGE CHARACTER. (CPU 212/214/216)
SM89	PORT 0: END OF MESSAGE CHARACTER. (CPU 212/214/216)
SM90 SM91	PORT 0: IDLE LINE TIME PERIOD GIVEN IN MILLISECONDS. THE FIRST CHARACTER RECEIVED AFTER THE IDLE LINE TIME HAS EXPIRED IS THE START OF A NEW MESSAGE. SM90 IS MSB. (CPU 212/214/216)
SM92 SM93	PORT 0: INTER-CHARACTER/MESSAGE TIMER TIMEOUT VALUE GIVEN IN MILLISECONDS. IF THE TIME PERIOD IS EXCEEDED, THE RECEIVE MESSAGE IS TERMINATED. SM92 IS MSB. (CPU 212/214/216)
SM94	PORT 0: MAXIMUM NUMBER OF CHARACTERS TO BE RECEIVED (1 TO 255 BYTES) (CPU 212/214/216)

FIG. 30D

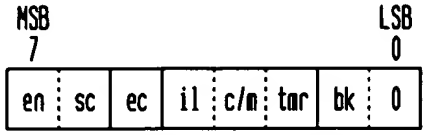
SM BIT DEFINITION (READ/WRITE) (CONTINUED)	
SM BITS	DESCRIPTION
SM187	PORT 1: MESSAGE INTERRUPT CONTROL BYTE, (CPU 216 ONLY)  en: ENABLE/DISABLE RECEIVE MESSAGE BIT IS CHECKED EACH TIME THE RCV INSTRUCTION IS EXECUTED. IF THIS BIT IS A "0", THEN THE RECEIVE MESSAGE FUNCTION IS DISABLED. IF THIS BIT IS A "1", THEN THE RECEIVE MESSAGE FUNCTION IS ENABLED. sc: 0 - IGNORE SMB188; 1 - USE THE VALUE OF SMB188 TO DETECT START OF MESSAGE ec: 0 - IGNORE SMB189; 1 - USE THE VALUE OF SMB189 TO DETECT END OF MESSAGE il: 0 - IGNORE SMW190; 1 - USE THE VALUE OF SMW190 TO DETECT AN IDLE CONDITION[1] c/m: 0 - USE TIMER AS AN INTER-CHARACTER TIMER; 1 - USE TIMER AS A MESSAGE TIMER tmr: 0 - IGNORE SMW192; 1 - TERMINATE RECEIVE IF THE TIME PREIOD IN SMW192 IS EXCEEDED bk: 0 - IGNORE BREAK CONDITIONS; 1 - USE BREAK CONDITION AS START OF MESSAGE DETECTION [1]BY SETTING THE sc AND bk BITS TO 0 AND THE en, il, c/m AND tmr BITS TO 1 WITH AN IDLE LINE TIMER VALUE OF ZERO, SMW192 WILL BE USED TO TIME OUT THE RCV INSTRUCTION WITHOUT RECEIVING ANY CHARACTERS. IF THE TIMER IS NOT USED (tmr = 0), THEN ANY CHARACTER RECEIVED WILL BE USED AS START OF MESSAGE. THE BITS OF THE MESSAGE INTERRUPT CONTROL BYTE ARE USED TO DEFINE THE CRITERIA BY WHICH THE MESSAGE IS IDENTIFIED. BOTH START OF MESSAGE AND END OF MESSAGE CRITERIA ARE DEFINED. TO DETERMINE THE START OF A MESSAGE EITHER OF TWO SETS OF LOGICALLY ANDED START OF MESSAGE CRITERIA MUST BE TRUE AND MUST OCCUR IN SEQUENCE (IDLE LINE FOLLOWED BY START CHARACTER OR BREAK FOLLOWED BY START CHARACTER). TO DETERMINE THE END OF A MESSAGE THE ENABLED END OF MESSAGE CRITERIA ARE LOGICALLY Ored. THE EQUATIONS FOR START AND STOP CRITERIA ARE GIVEN BELOW: Start of Message = il * sc + bk * sc End of Message = ec + tmr + maximum character count reached NOTE: RECEIVE WILL AUTOMATICALLY BE TERMINATED BY AN OVERRUN OR A PARITY ERROR (IF ENABLED).
SM188	PORT 1: START OF MESSAGE CHARACTER. (CPU 216 ONLY)
SM189	PORT 1: END OF MESSAGE CHARACTER. (CPU 216 ONLY)
SM190 SM191	PORT 1: IDLE LINE TIME PERIOD GIVEN IN MILLISECONDS. THE FIRST CHARACTER RECEIVED AFTER THE IDLE LINE TIME HAS EXPIRED IS THE START OF A NEW MESSAGE. SM190 IS MSB. (CPU 216 ONLY)
SM192 SM193	PORT 1: INTER-CHARACTER/MESSAGE TIMER TIMEOUT VALUE GIVEN IN MILLISECONDS. IF THE TIME PERIOD IS EXCEEDED, THE RECEIVE MESSAGE IS TERMINATED. SM192 IS MSB. (CPU 216 ONLY)
SM194	PORT 1: MAXIMUM NUMBER OF CHARACTERS TO BE RECEIVED (1 TO 255 BYTES) (CPU 216 ONLY)

FIG. 30E

PLC TYPE	PORT 0			PORT 1		
	NUMBER OF CONNECTIONS		BUFFER SIZE	NUMBER OF CONNECTIONS		BUFFER SIZE
	TOTAL	RESERVED	(BYTES)	TOTAL	RESERVED	(BYTES)
CPU 221	4	1 - PG 1 - OP	128/256	-	-	-
CPU 222	4	1 - PG 1 - OP	128/256	-	-	-
CPU 224	4	1 - PG 1 - OP	128/256	-	-	-
CPU 226	4	1 - PG 1 - OP	128/256	4	1 - PG 1 - OP	128/256

FIG. 31

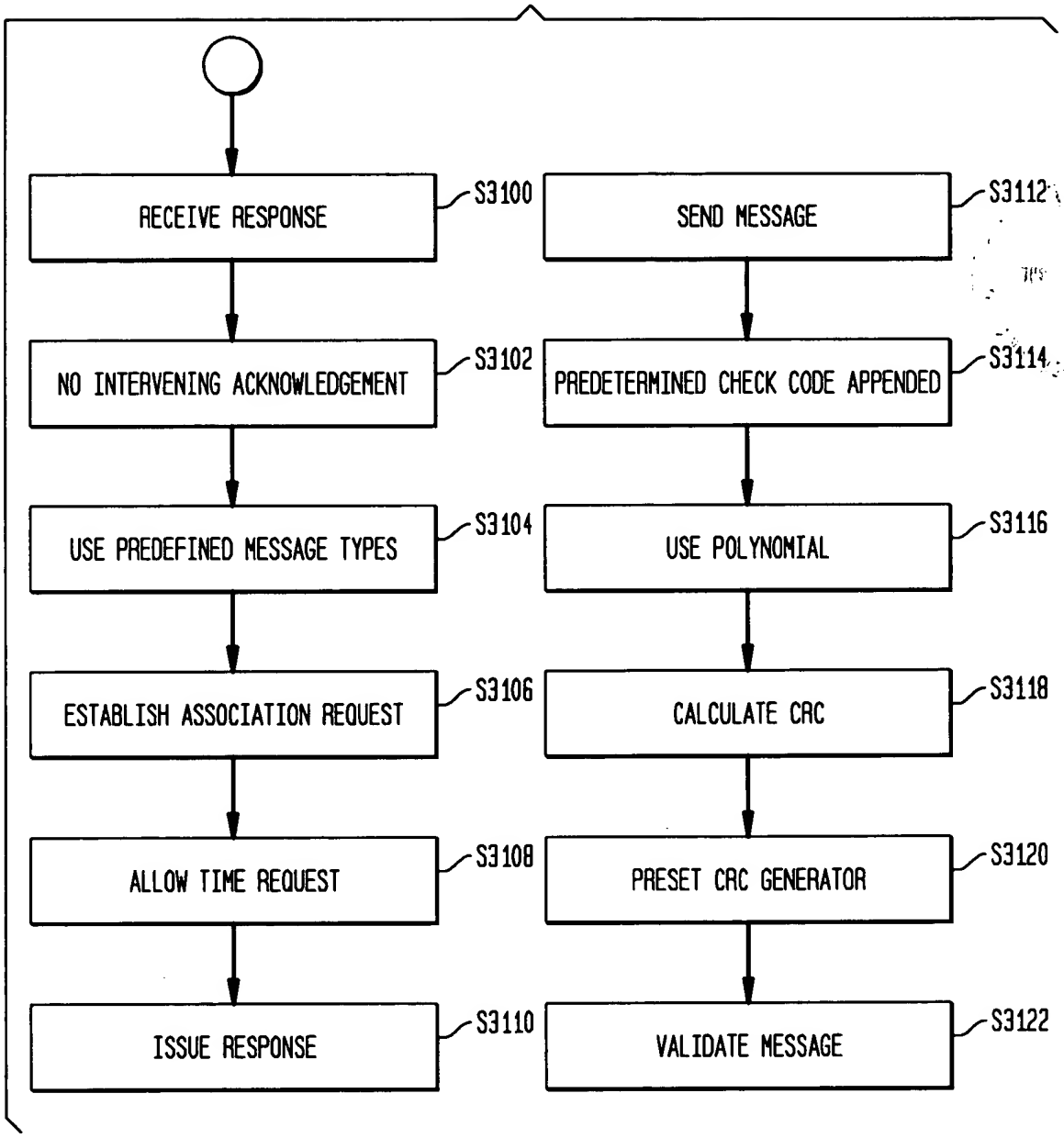

RECEIVE RESPONSE
S3100
NO INTERVENING ACKNOWLEDGEMENT
S3102
USE PREDEFINED MESSAGE TYPES
S3104
ESTABLISH ASSOCIATION REQUEST
S3106
ALLOW TIME REQUEST
S3108
ISSUE RESPONSE
S3110
SEND MESSAGE
S3112
PREDETERMINED CHECK CODE APPENDED
S3114
USE POLYNOMIAL
S3116
CALCULATE CRC
S3118
PRESET CRC GENERATOR
S3120
VALIDATE MESSAGE
S3122

FIG. 32A

PROGRAM CONTROL FUNCTIONS				
INSTRUCTION	MNEMONIC & OPERAND(S)	DESCRIPTION	STL STATUS ELEMENT	VALID OPERANDS
HIDE	HIDE n, a, ρ	THE HIDE LABEL MARKS THE START OF A BLOCK OF n INSTRUCTIONS THAT ARE ENCRYPTED USING THE PASSWORD, ρ, WHEN a IS NON-ZERO.	STK, SMB1	ENABLE: NONE n: KW(2 BYTES) (UI) a: KW(2 BYTES) (UI) ρ: KD(4 BYTES) (UI)

FIG. 32B

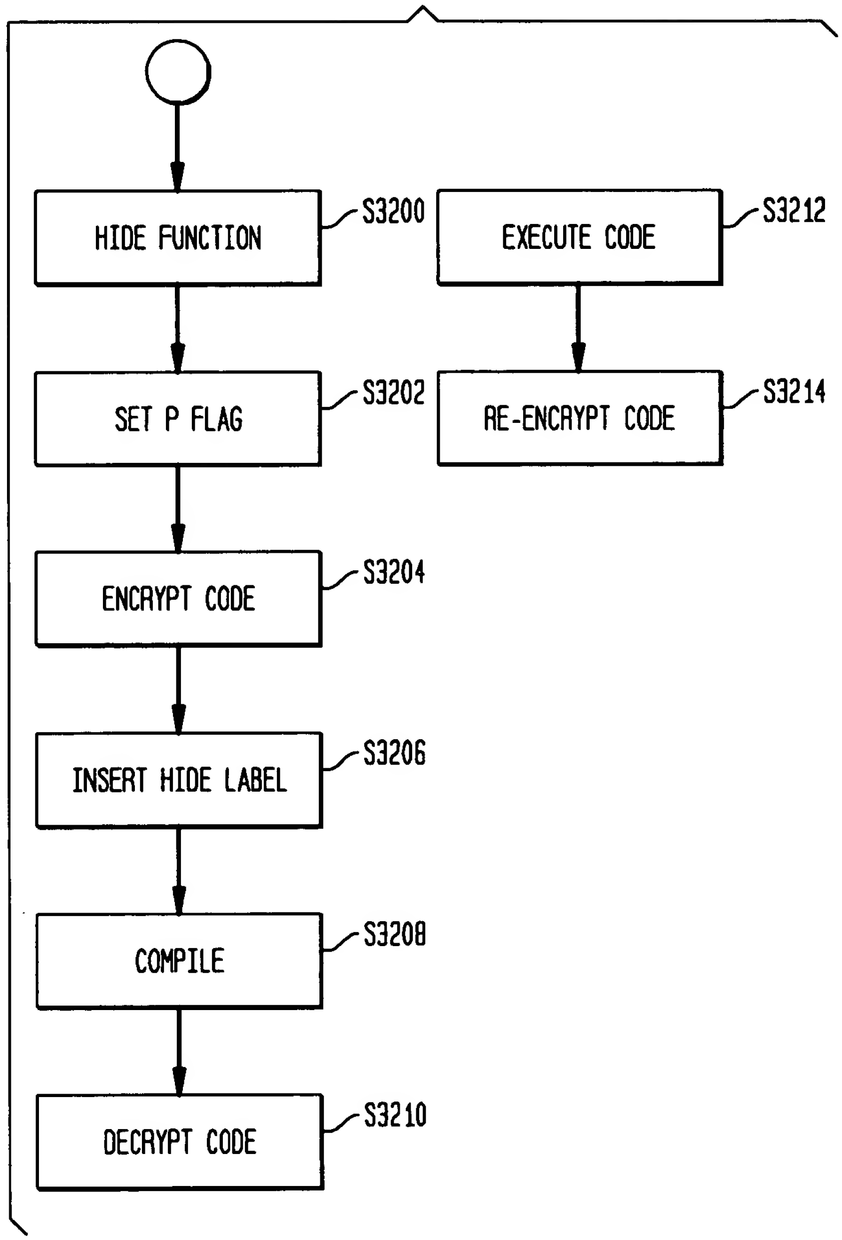


FIG. 33A

PROGRAM CONTROL FUNCTIONS				
INSTRUCTION	MNEMONIC & OPERAND(S)	DESCRIPTION	STL STATUS ELEMENT	VALID OPERANDS
SYSTEM FUNCTION CALL	SFC f,s, #_parms, p0,p1,p2,...	WHEN S0 = 1, EXECUTE THE SYSTEM FUNTION IDENTIFIED BY f AND THE SUB-FUNCTION IDENTIFIED BY s.	STK, SMB1, <p0>, <p1>, <p2>,...	ENABLE: S0 f: KW (UI) 0-65536 s: KW (UI) 0-65536 #parms: KB (UI) 0-16 p0,...: Bit,Byte,Word, Dword Bit V,I,Q,M,SM,S, T,C,L Byte VB,IB,QB,MB, SMB,SB,KB,LB Word VW,T,C,IW, QW,MW,SMW, SW,LW,KW Dword VD,ID,QD,MD, SMD,SD,HC, LD,KD,&VB, &IB,&QB,&MB, &T,&C,&SB NOTE: CONSTANTS AND ADDRESS POINTER SPECIFICATIONS ARE NOT ALLOWED FOR OUTPUT OR INPUT/OUTPUT PARAMETERS.

FIG. 33B

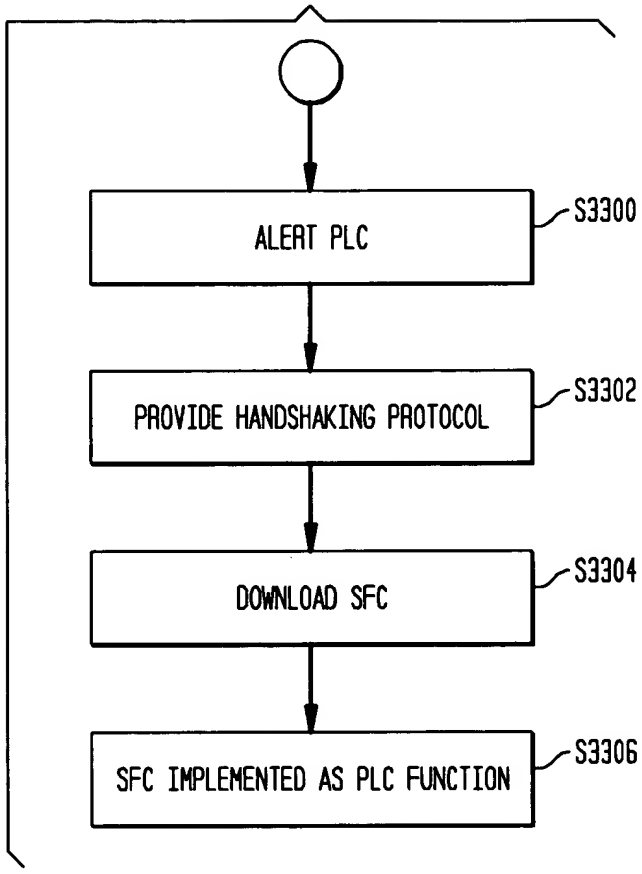

ALERT PLC
S3300
PROVIDE HANDSHAKING PROTOCOL
S3302
DOWNLOAD SFC
S3304
SFC IMPLEMENTED AS PLC FUNCTION
S3306

FIG. 34A

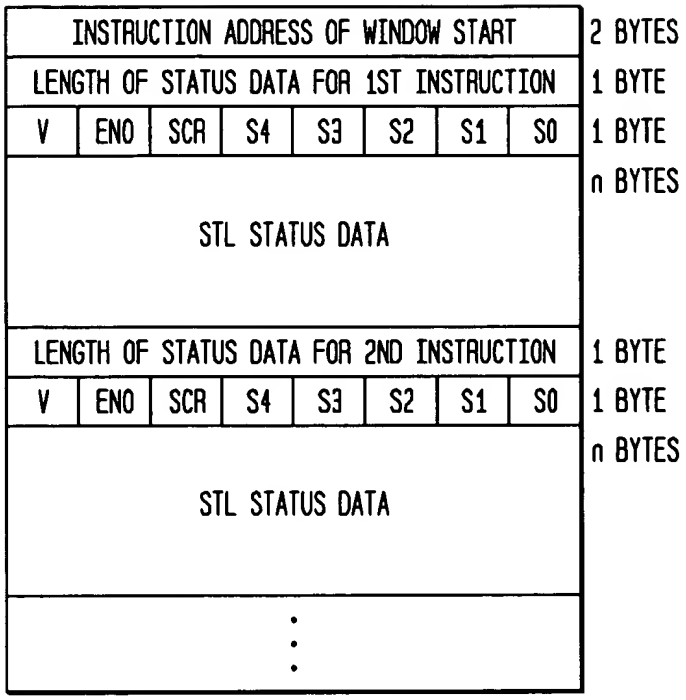


FIG. 34B

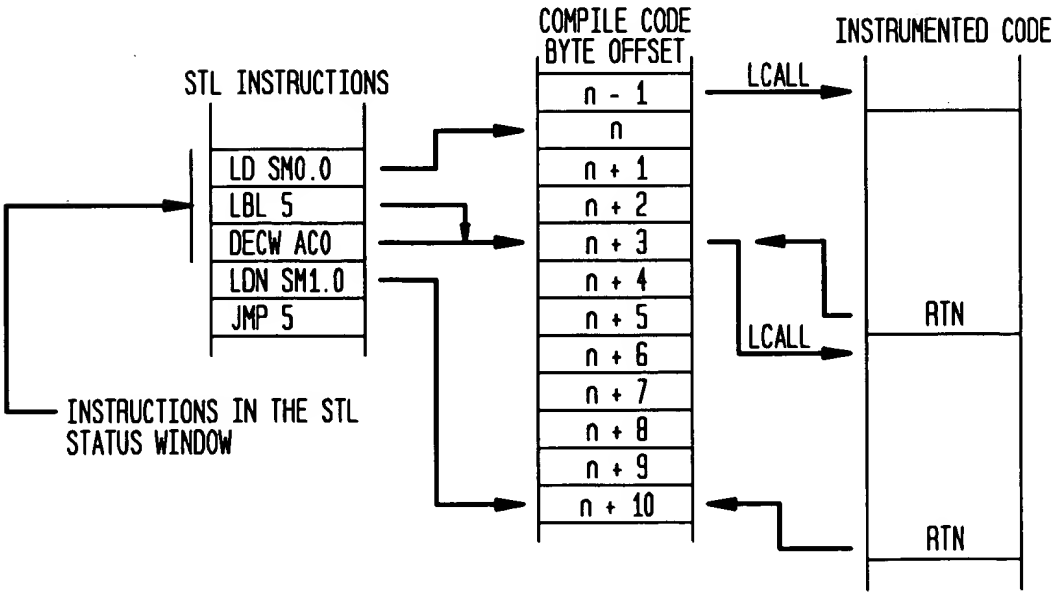


FIG. 34C

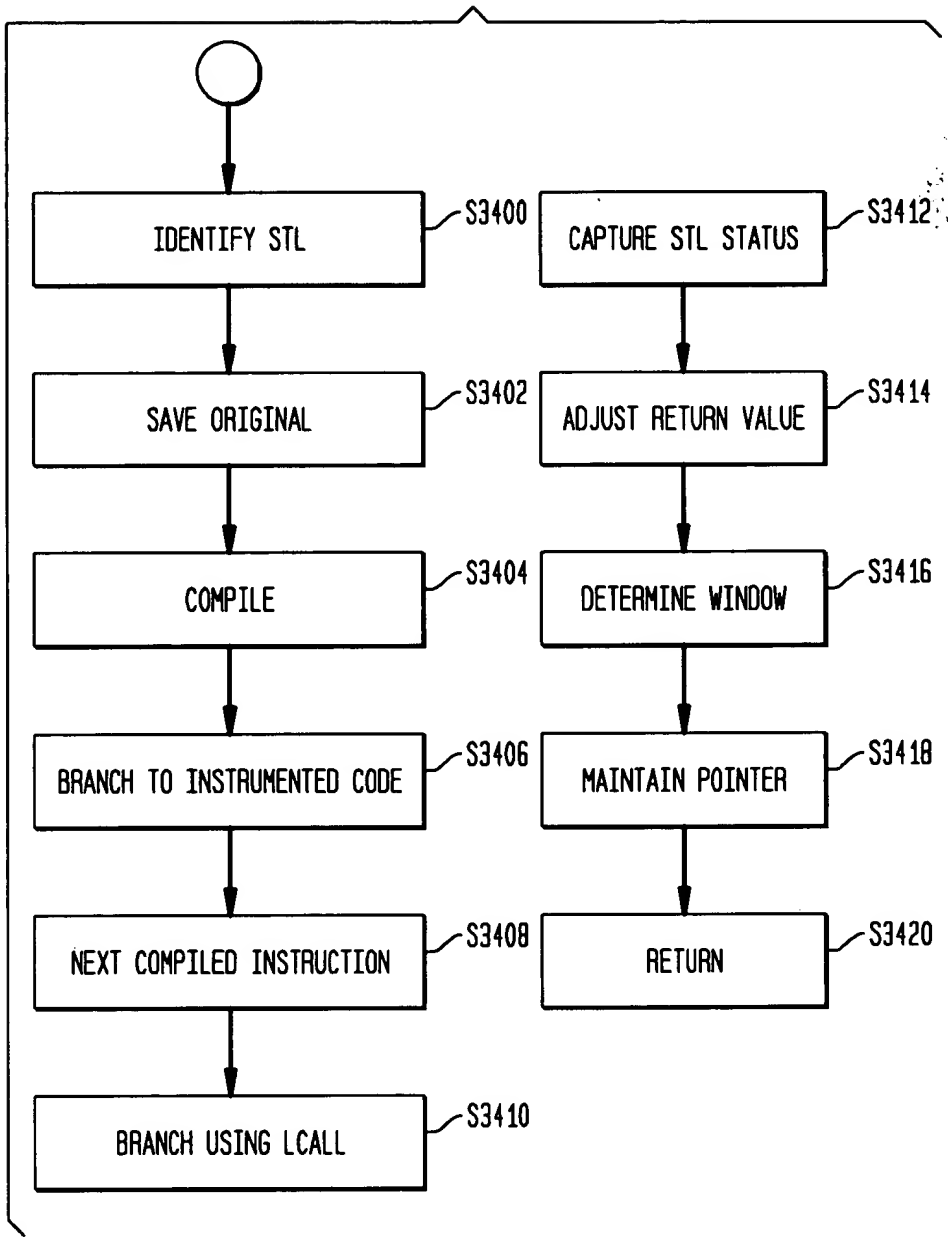


FIG. 34D

	BASIC INSTR.	COMPILED TO	BYTES	CYCLES
1	LD	RLC A MOV C,<BIT>	1 2	1 1
2	LDN	RLC A MOV C,<BIT> CPL C	1 2 1	1 1 1
3	A	ANL C,<BIT> NOP	2 1	2 1
4	AN	ANL C,/<BIT> NOP	2 1	2 1
5	O	ORL C,<BIT> NOP	2 1	2 1
6	ON	ORL C,/<BIT> NOP	2 1	2 1
7	=	MOV <BIT>,C NOP	2 1	2 1
8	ALD	ANL C,ACC.0 RR A	2 1	2 1
9	OLD	ORL C,ACC.0 RR A	2 1	2 1
10	NOT	CPL C NOP NOP	1 1 1	1 1 1
11	LPS	RL A MOV ACC.0,A	1 2	1 2
12	LPP	RRC A NOP NOP	1 1 1	1 1 1
13	LRD	MOV C,ACC.0	2	2
14	RET	RET NOP NOP	1 1 1	2 1 1
15	INT	CLR A NOP NOP	1 1 1	1 1 1
16	END	JNC $+0 RET	2 1	2 2
17	CRETI	JNC $+0 RET	2 1	2 2

FIG. 35A

HOUSING	HEIGHT	WIDTH	DEPTH
CPU 221	80 mm	90 mm	62 mm
CPU 222	80 mm	90 mm	62 mm
CPU 224	80 mm	120.5 mm	62 mm
CPU 226	80 mm	190 mm	62 mm
8 POINT I/O MODULE	80 mm	46 mmm	62 mm
16 POINT I/O MODULE	80 mm	71.2 mm	62 mm
32 POINT I/O MODULE	80 mm	125 mm	62 mm
INTELLIGENT MODULE	80 mm	90 mm	62 mm

FIG. 35B

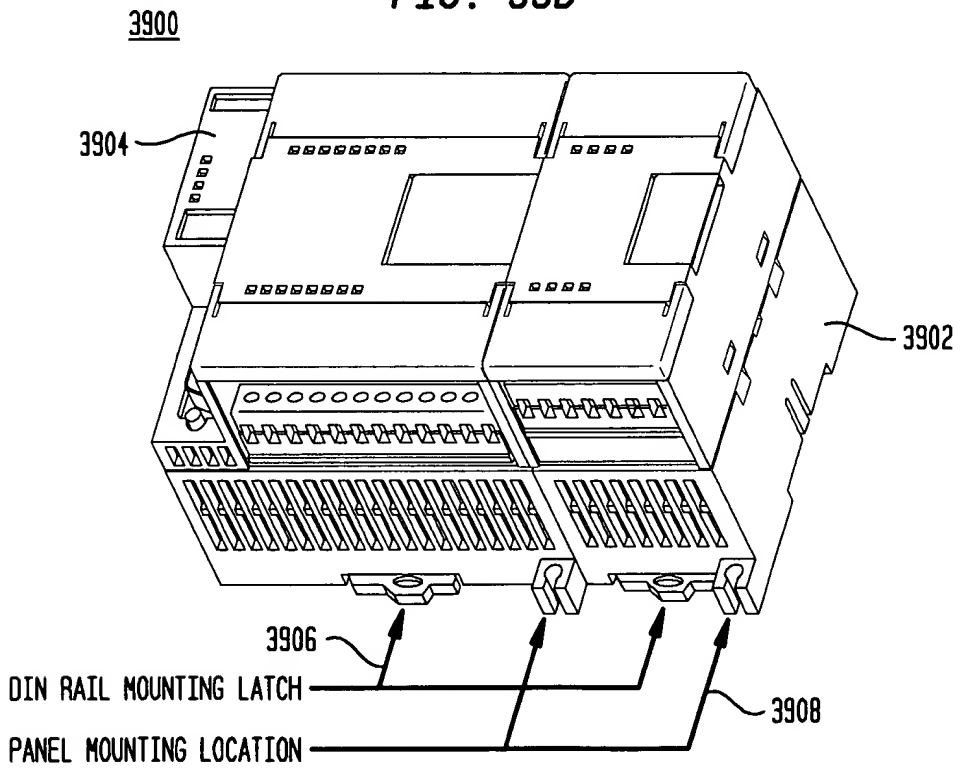


FIG. 36

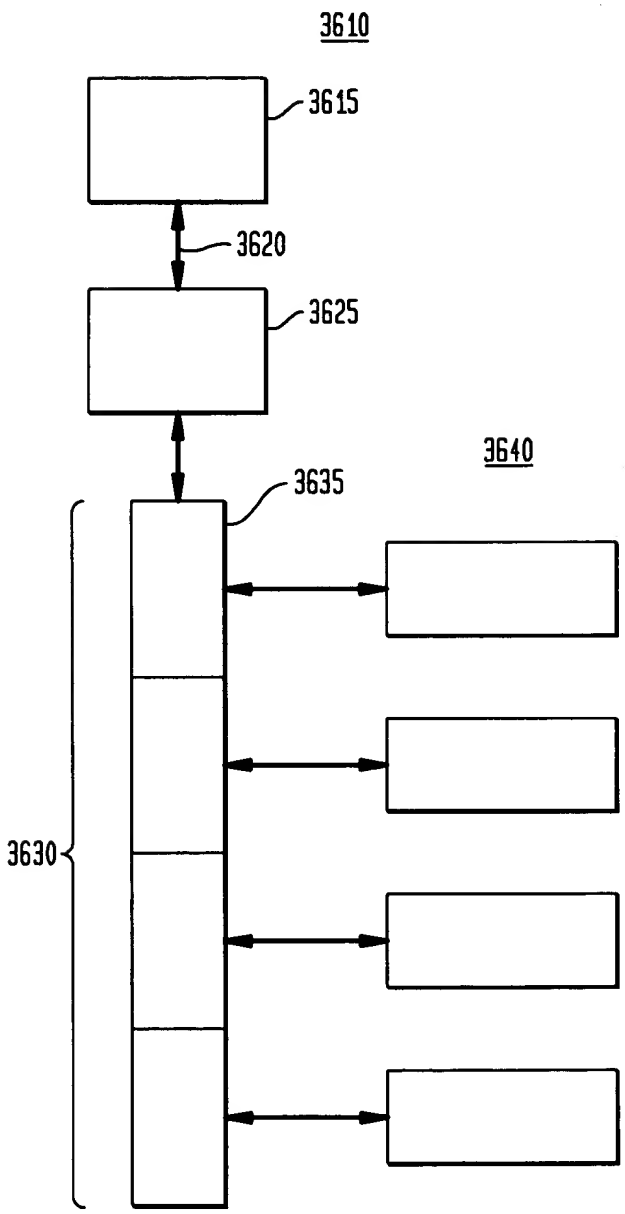


FIG. 37A

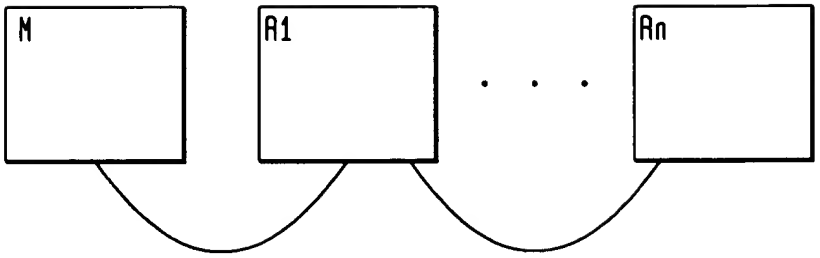


FIG. 37B

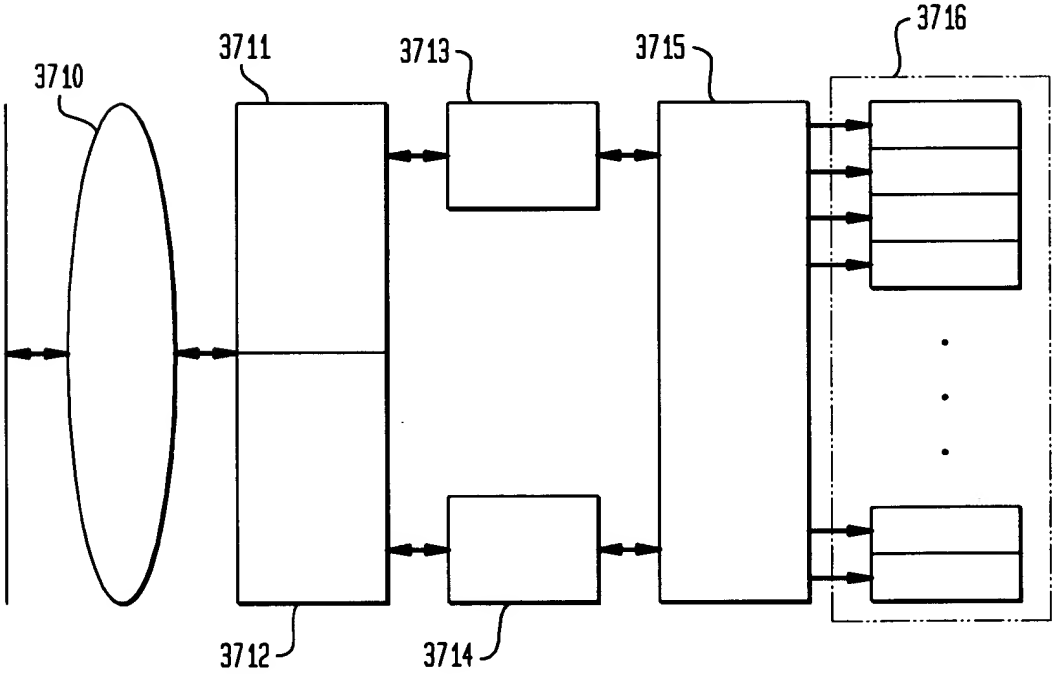


FIG. 38

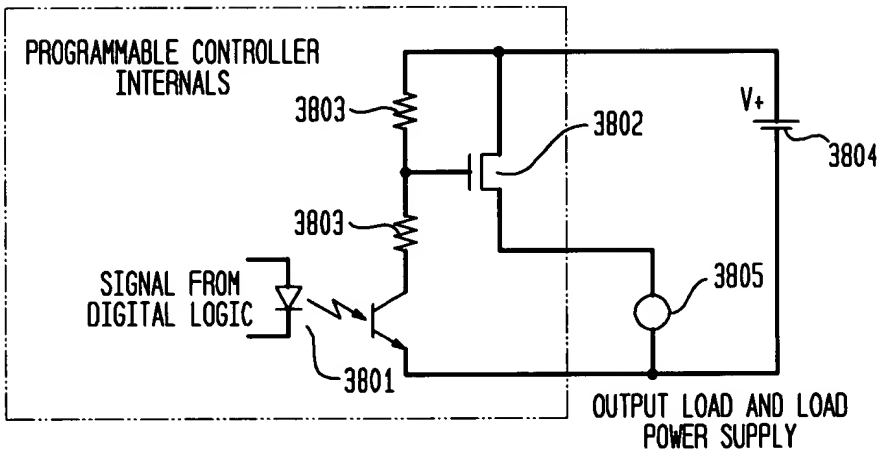
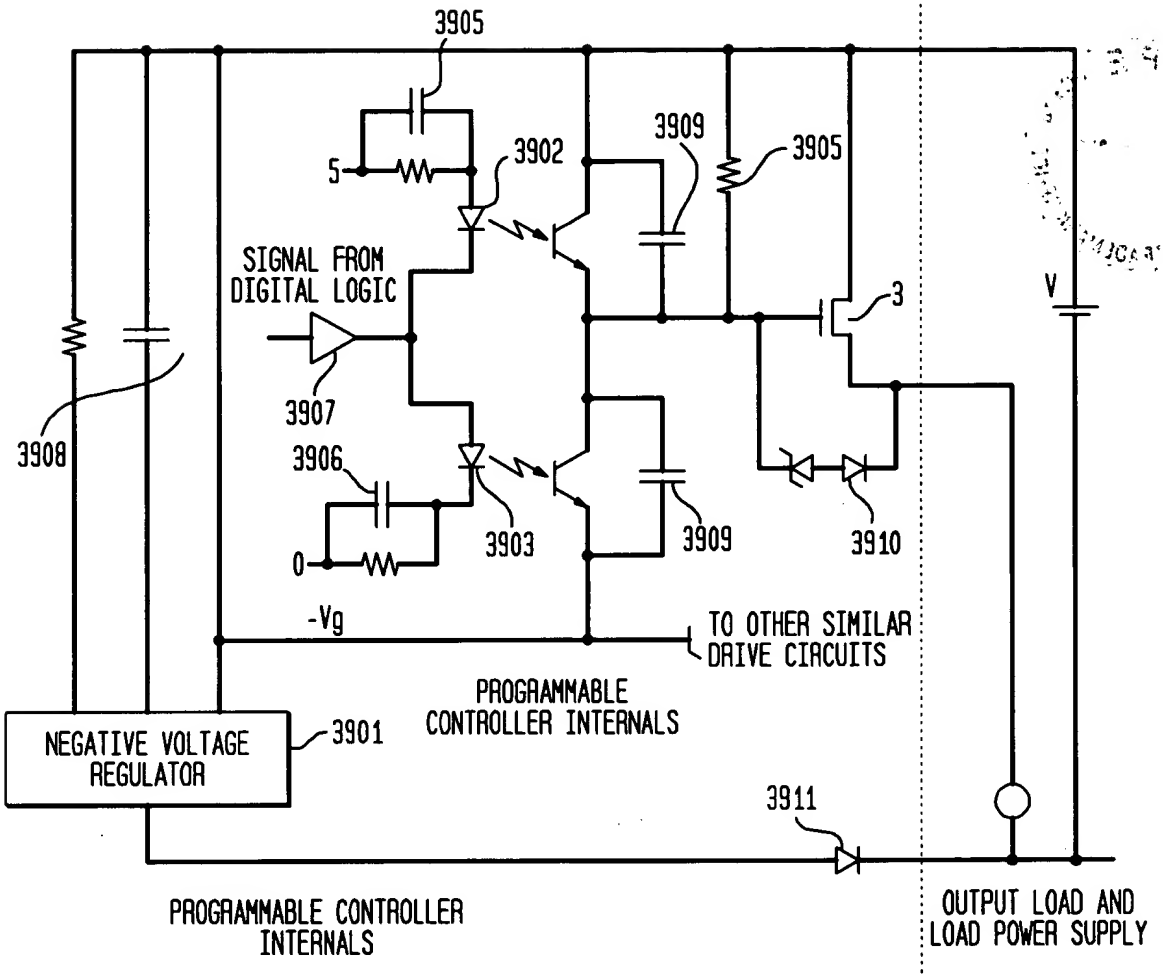


FIG. 39


3905
5
3902
3909
3905
SIGNAL FROM
DIGITAL LOGIC
3
V
3907
3908
3906
0
3903
3909
3910
-Vg
TO OTHER SIMILAR
DRIVE CIRCUITS
PROGRAMMABLE
CONTROLLER INTERNALS
NEGATIVE VOLTAGE
REGULATOR
3901
3911
PROGRAMMABLE CONTROLLER
INTERNALS
OUTPUT LOAD AND
LOAD POWER SUPPLY